THE UNIVERSITY
of EDINBURGH

BUILDING MODELS FROM MULTIPLE POINT SETS WITH KERNEL DENSITY ESTIMATION

STEVEN MCDONAGH

Doctor of Philosophy

Institute of Perception, Action and Behaviour
School of Informatics
University of Edinburgh

2015

To Mum and Dad.

— Steven

LAY SUMMARY

The point set registration family of techniques involve automatically and accurately aligning sets of matching points in space. These techniques are used to help solve many computer vision related problems and consequently find usage in many important applications. In particular, point set registration can be considered one of the crucial stages involved in the digital reconstruction of models of physical scenes and objects from real-world depth sensor measurements. Object shape measurements, recorded from varying points of view, result in multiple disparate point sets. The problem of aligning these multiple point sets must be solved before full, watertight model reconstruction can be performed. This constitutes a complex task that is imperative due to the large number of critical functions that accurate and reliable full model reconstructions contribute to.

In this thesis we improve the quality and feasibility of model and environment reconstruction through the enhancement of multi-view point set registration techniques. The thesis makes the following contributions: First, we demonstrate that employing robust surface inference techniques to reason about the real-world surfaces that range sensors measure allow us to mitigate measurement uncertainty and also to separate the problems of model design and viewpoint alignment optimisation. Our surface estimates provide a novel quality metric with which to inform the point set registration process and thus aid view alignment. By estimating surfaces directly from sets of points and performing experiments on a variety of point datasets we demonstrate that we have developed an effective solution to the simultaneous multi-view registration problem.

We then focus on constructing a distributed computation framework capable of solving high-throughput computational problems. We present a novel computational model that we call Semi-Synchronised Task Farming (SSTF), capable of modelling and subsequently solving computationally distributable problems that benefit from both independent and dependent distributed components and a level of communication between process elements. We demonstrate that this framework is a novel schema for parallel computer vision algorithms and evaluate the performance to establish computational gains over serial implementations. We couple this framework with an accurate pre-

diction model to provide a novel distributed-computation-time inference tool. This framework proves appropriate for instantiating expensive real-world algorithms with substantial parallel performance gains and predictable time savings.

Finally, we focus on a timely instance of the multi-view registration problem: modern range sensors provide large numbers of viewpoint samples that result in an abundance of depth data information. The ability to utilise this abundance of depth data in a feasible and principled fashion is of importance to many emerging application areas making use of spatial information. We develop novel methodology for the registration of depth measurements acquired from many viewpoints capturing physical object surfaces. By defining registration and alignment quality metrics based on our surface inference framework we construct an optimisation methodology that implicitly considers all viewpoints simultaneously. We use a data-driven approach to consider varying object complexity and guide large view-set alignment. By aligning large numbers of partial, arbitrary-pose views we evaluate this strategy quantitatively on large view-set range sensor data where we find that we can improve registration accuracy over existing methods and contribute increased registration robustness to initial misalignment. This allows large-scale registration on problem instances exhibiting varying object complexity with the added advantage of massive parallel efficiency.

In summary, we propose novel view alignment methodology and practical routes to solving all stages of the process when tackling large sets of sensor measurements representing varying viewpoints of physical objects.

ABSTRACT

One of the fundamental problems in computer vision is point set registration. Point set registration finds use in many important applications and in particular can be considered one of the crucial stages involved in the reconstruction of models of physical objects and environments from depth sensor data. The problem of globally aligning multiple point sets, representing spatial shape measurements from varying sensor viewpoints, into a common frame of reference is a complex task that is imperative due to the large number of critical functions that accurate and reliable model reconstructions contribute to.

In this thesis we focus on improving the quality and feasibility of model and environment reconstruction through the enhancement of multi-view point set registration techniques. The thesis makes the following contributions: First, we demonstrate that employing kernel density estimation to reason about the unknown generating surfaces that range sensors measure allows us to express measurement variability, uncertainty and also to separate the problems of model design and viewpoint alignment optimisation. Our surface estimates define novel view alignment objective functions that inform the registration process. Our surfaces can be estimated from point clouds in a data-driven fashion. Through experiments on a variety of datasets we demonstrate that we have developed a novel and effective solution to the simultaneous multi-view registration problem.

We then focus on constructing a distributed computation framework capable of solving generic high-throughput computational problems. We present a novel task-farming model that we call Semi-Synchronised Task Farming (SSTF), capable of modelling and subsequently solving computationally distributable problems that benefit from both independent and dependent distributed components and a level of communication between process elements. We demonstrate that this framework is a novel schema for parallel computer vision algorithms and evaluate the performance to establish computational gains over serial implementations. We couple this framework with an accurate computation-time prediction model to contribute a novel structure appropriate for

addressing expensive real-world algorithms with substantial parallel performance and predictable time savings.

Finally, we focus on a timely instance of the multi-view registration problem: modern range sensors provide large numbers of viewpoint samples that result in an abundance of depth data information. The ability to utilise this abundance of depth data in a feasible and principled fashion is of importance to many emerging application areas making use of spatial information. We develop novel methodology for the registration of depth measurements acquired from many viewpoints capturing physical object surfaces. By defining registration and alignment quality metrics based on our density estimation framework we construct an optimisation methodology that implicitly considers all viewpoints simultaneously. We use a non-parametric data-driven approach to consider varying object complexity and guide large view-set spatial transform optimisations. By aligning large numbers of partial, arbitrary-pose views we evaluate this strategy quantitatively on large view-set range sensor data where we find that we can improve registration accuracy over existing methods and contribute increased registration robustness to the magnitude of coarse seed alignment. This allows large-scale registration on problem instances exhibiting varying object complexity with the added advantage of massive parallel efficiency.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor, Bob Fisher, for the continuous and substantial time, effort and thought he has invested in my graduate experience. The combination of his expertise, encouragement, patience and trust contributed immensely to the excellent research environment I have been lucky enough to be a part of for the last five years.

During this time, I was very fortunate to collaborate closely with Bastiaan Boom, Cigdem Beyan, Phoenix Huang, Peter Sandilands, Sergio Orts-Escolano and Helen Ramsden. The quality of this work would be much the poorer without these colleagues and companions. I am also grateful to my supervisory panel members; Subramanian Ramamoorthy and Amos Storkey for their continued guidance and support.

I thank the Institute of Perception, Action and Behaviour staff and students for their involvement and countless interactions. It has been a pleasure to exchange ideas with everyone in IPAB and the wider School of Informatics where I have had the opportunity to undertake a PhD in computer vision, with few restrictions placed on my research.

Numerous friends and family members have offered me their support, advice and good humour throughout my studies, too many to list here. I thank them all. In particular, there are countless reasons to thank my parents, Mike and Jane McDonagh, as well as my sister Rose McDonagh. Without their continued support, encouragement and guidance this thesis would simply not exist.

DECLARATION

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

The material presented in this thesis has contributed to published work. I hereby declare my author contributions to the related publications:

Simultaneous registration of multi-view range images with adaptive kernel density estimation. S. McDonagh and R. B. Fisher. *14th IMA Conference on Mathematics of Surfaces*. pp 31–62, Birmingham, 2013 [176].

Conceived and designed the methodology; conceived and designed the experiments; performed the experiments; analysed the data; contributed materials, analysis tools; wrote the paper.

Applying semi-synchronised task farming to large-scale computer vision problems. S. McDonagh, C. Beyan, P. X. Huang and R. B. Fisher. *International Journal of High Performance Computing Applications*, 2014 [178].

Conceived and designed the methodology in conjunction with second, third authors; conceived and designed the experiments in conjunction with second, third authors; performed the experiments in conjunction with second, third authors; collectively analysed the data; contributed materials, analysis tools; co-wrote the paper.

Laminar and Dorsoventral Molecular Organization of the Medial Entorhinal Cortex Revealed by Large-scale Anatomical Analysis of Gene Expression. H. L. Ramsden, G. Sürmeli, S. McDonagh and M. F. Nolan. *PLOS Computational Biology*, 2015 [212].

Conceived and designed the methodology in conjunction with first author; performed the experiments in conjunction with first author; contributed analysis tools.

Edinburgh, 2015

Steven McDonagh

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Part I

INTRODUCTION

1

INTRODUCTION

A fundamental objective of computer vision involves simulating the human visual system and advancing the theory underlying artificial systems capable of extracting meaningful information from light sensors and images. Biological visual perception plays a hugely significant role in allowing humans to understand, interact with and navigate in their surroundings. Humans are highly capable of performing natural tasks such as recognising a familiar person, manipulating physical objects and moving within their environment. These examples provide a selection of the many complex tasks that still present difficult challenges for computers and autonomous systems. To illustrate why these remain challenging, the type of visual system we aim to simulate should first be well defined. A visual system is a collection of devices that transform measurements of light into information about spatial and material properties of a scene. Such a system contains visual sensors such as eyes in the case of the human or digital sensors (*e.g.* cameras) in the case of the computer, and computational units such as the brain for the human or the CPU for the digital system. While these sensors record the intensity of light that hits the photosensitive cells (pixels in the case of digital cameras), the computational units decipher the measured values to infer the characteristics of the scene that is being observed.

We require models to interpret the captured measurements of light and to then derive meaningful information from them. Such models are generally a simplified representation of the physical world. Relevant model classes include image formation models that attempt to provide good explanations for the *appearance* of object surfaces and 3D models that aim to provide a *geometrical* understanding of the perceived scene. It is in this sense that building good 3D models of an observed scene provides a set of important challenges for computer vision. A large body of work and research has been

carried out on this topic stretching decades, and continuing in recent years, however the task constitutes a set of challenging problems that remain to be completely solved.

Depth information and global object *shape* can be acquired using depth sensing systems and 3D scanners in a similar way that reflectance, illumination and positional properties can be captured utilising 2D imaging devices and object movement can be measured with motion capture devices. This thesis focuses on challenges relating to the former of these, namely acquiring and measuring the 3D *shape* of real-world objects using measurements of the real world obtained from depth sensors.

3D vision and perception is difficult even for humans in some instances. This can be observed by *e.g.* considering the many well known examples of 3D optical illusions and related visual phenomena. Even the healthy adult human brain is capable of making mistakes and incorrect inferences relating to what is being perceived in a surrounding 3D environment. This reinforces the fact that designing computational 3D vision (*i.e.* constructing algorithms that allow the computer to perceive the 3D world as a human would) requires solving exceedingly difficult sets of interrelated problems. The motivation to undertake these challenges and the merit of finding good solutions is however substantial. The topic of 3D vision is well reviewed in [165] with an overview of practical computer vision based applications provided by [264]. Modern and thorough treatments of computer vision topics are given by several authors (see *e.g.* [96, 251, 93]).

1.1 VISION AND 3D MODELLING

Two dimensional images alone are inadequate to understand the full complexity of our environment. Reasoning about the 3D geometrical information contained in an environment allows for more complete explanations of objects that exist in a scene and a more detailed understanding of components that objects consist of. It is well understood that observing objects from varying viewpoints or illumination can change object *appearance* subtly or indeed drastically. Using only 2D image observations, it can be difficult to accurately obtain global and local object attributes such as position, size, shape and additional fine detail. However, in the case of humans, by simply handling or touching an object it often becomes possible to infer 3D object *shape* and facilitate more complex tasks *e.g.* object recognition and classification. The addition of simple tactile modes often allow humans to naturally conceptualise and infer a 3D model

(object representation) purely from the sensing of shape. It is by combining both visual observations and 3D object models that humans are able to interpret their physical surroundings. For both natural and autonomous visual systems to be efficient, it is thus essential to have 3D models of the environment.

The pipeline that produces 3D models from images is often divided into the steps of: (1) *information extraction* involving the extraction of 3D information from 2D observations. Extracted information typically takes the form of 3D points in space (*coordinates*) or surface normals and (2) *modelling* where the task is to fuse all previously extracted 3D information into a compact 3D representation (model) of observed objects or scenes. Digital 3D models typically take the form of *e.g.* meshes, clouds of points, a 3D grid of voxels or implicit surfaces. Algorithmic pipelines used to produce such 3D models have proved successful and useful tools in a large and varied selection of application areas. Examples include:

- Reverse engineering
- Face and gesture recognition
- e-Heritage
- Industrial quality control
- Autonomous vehicles
- Medical image analysis

To briefly characterise these examples; reverse engineering of object shape allows the measurement and analysis of geometric form enabling tasks such as distance measurement, symmetry checking and deformation control. By using 3D models to understand the structure of objects, improved production quality can be obtained [92, 269]. Improvements have also been made to the quality and robustness of face and gesture recognition when dealing with complex backgrounds and changes in appearance by utilising object *shape* (*e.g.* [1]). The additional application of e-Heritage [20, 132] involves digitally preserving objects of important cultural heritage value. The loss and deterioration of valued historical objects can be mitigated by capturing object shape and enabling digital preservation in areas where objects are deteriorating or facing destruction due to *e.g.* natural weathering, disaster or war.

The further example application area of quality control involves *e.g.* minimising construction defects. Such defects, experienced during object construction, are often costly yet preventable. 3D models are now commonly utilised for expansion and renovation projects in many construction sectors and quality control [6]. The tasks of spatial reasoning for *e.g.* autonomous driving and vision in hostile environments [46, 47] have additionally received much attention of late, especially from the robotics community with organisations and competitions such as the DARPA grand challenges [38], spurring research in the field. Sensing and localisation capabilities constitute one of the main challenges in this domain and therefore reconstructing accurate environment geometry is again of key importance.

In medical imaging domains, 3D models are able to aid many image analysis tasks and have proven to be effective in segmenting, tracking, matching and classifying anatomic structure (*e.g.* [177]). Furthermore, if accurate 3D models can be constructed in this domain, they are able to support intuitive interaction, allowing medical scientists and practitioners to exercise their image interpretation expertise enabled by additional spatial information and models [179].

In summary there are many contemporary application areas where successfully building and reasoning about geometric models is a driver of novel work and are of prime importance. While such modelling techniques prove useful, accurately constructing a digital 3D scene or object by hand is both time consuming and difficult. Automating the 3D modelling pipeline is of key significance for the computer vision community and has accordingly attracted much interest that we will go on to survey in detail, in Chapter 2.

A central component of automating the digital 3D object reconstruction pipeline, for use in the highlighted fields, is the automated fusion of the extracted 3D information into compact models. In practice this typically involves solving the global *registration* problem, the alignment of all sensor viewpoints into a common frame of reference (see section 1.3). Contemporary depth sensors are able to rapidly provide orders of magnitude more depth information than has traditionally been considered, making optimising this fusion process challenging. In this thesis we claim that utilising this recent abundance of available data is however important and beneficial (*e.g.* in terms of model quality, completeness, accuracy). Accordingly, a key contribution of this work is the methodology for solving the global registration problem whilst considering modern,

large depth measurement datasets and investigate the benefits afforded by fusing such large sets of sensor views. We claim that solving the global registration problem for large view-sets in an accurate and feasible manner is both coveted and an obligatory requirement of modern modelling pipelines, owing in part to the progress of state-of-the-art depth measurement hardware. This thesis therefore provides valuable contributions to addressing problems that arise in the registration of large sets of point clouds. In the following sections we briefly outline the synthetic 3D modelling pipeline (1.2 - 1.3 and then formalise the thesis hypothesis and contributions (1.4 - 1.5).

1.2 SYNTHETIC 3D MODELLING

As discussed, obtaining accurate 3D models with synthetic systems is a highly challenging task relative to biological counterparts (*e.g.* the human eye). A 2D projection of an observed scene, recorded using a single digital camera, does not provide enough information to infer metrically accurate 3D geometry in the general case. To successfully extract a 3D surface; multiple samples from varying viewpoints are needed. By observing the same 3D spatial point from varying viewing positions, it becomes possible to retrieve its 3D coordinates. From a set of observations, it is therefore possible to reconstruct the 3D surface corresponding to overlapping regions between 2D images. Using multiple viewing positions is a popular technique and inferring 3D coordinates to carry out the *information extraction* step of the considered pipeline is well studied under the area of *multi-view geometry* [120].

Additionally recent progress in consumer-grade, direct depth acquisition devices such as structured light approaches, commodity depth cameras and LiDAR sensors are able to generate large view-set depth data. Consumer-grade range cameras have thus become a suitable data source for creating digital 3D models of physical objects. Range cameras are capable of generating large view-set, 3D point clouds that constitute popular scene or object representations in the vision and robotics communities for *e.g.* scene reconstruction and SLAM applications [135, 190, 76, 121, 261, 86]. Sensors are typically low cost, fast and possess an acceptable level of accuracy for many tasks. Contemporary examples of depth sensors in this class include the Microsoft Kinect [183], PrimeSense Carmine [209] and Asus Xtion Pro Live [10]. These sensors are able to provide high frame rate data streams that potentially result in large *view-set registration* problem

instances. In particular, such acquisition devices are typically able to return both depth data and colour images of scene or object, retrieving both the 3D shape and a colour image of an object from a fixed viewpoint. The acquired 3D image in this case is commonly known as a range image. The *information extraction* step (using multi-view geometry) can be omitted in such cases. However, from a single scene viewpoint, parts of the scene are occluded (possibly *self-occluded*) due to such sensors lacking an omnipresence. From this single viewpoint it is only possible to acquire a part of the scene or object. Therefore, multiple range images, acquired from various positions are needed to acquire all parts of the scene or object. In addition to consumer grade structured light and time-of-flight cameras, the quality of alternative technologies such as 3D laser scanning can also produce very dense high quality 3D point cloud data. Particularly if metric accuracy is an important property of the captured data, depth acquisition typically makes use of active optical devices. Such sensors are able to acquire high quality and dense point sets captured in the sensor field of view (FOV).

A range image, acquired by any of the studied acquisition methods, is obtained in a local coordinate system from the individual sensor viewpoint. Full models often require the collection of dozens or hundreds of views in order to build complete 3D models of the object or scene of interest. When starting from a set of independent views (that each lie in their own local reference frame) and, holding the hypothesis that sufficient view overlap and object surface coverage exists, it is possible to obtain a 3D model through the introduced *modelling* pipeline that can be partitioned into view *registration*, *integration* and surface *reconstruction* steps.

Active and passive depth sensors only measure the visible surface of a target scene and therefore only provide a partial view of an entity due to (self)-occlusions, blind areas or otherwise missing data. The reconstruction of complete and accurate models of physical entities from depth-sensor measurements therefore requires data from multiple viewpoints such that sufficient information can be acquired to minimise occluded areas and to redundantly measure surface detail in an effort to average out errors found in individual frames.

As we note, constructing such models from partial views typically involves the fusion and global *registration* (alignment) of sensor viewpoints into a common reference frame. The point set registration task is a fundamental problem in computer vision and in particular, forms a crucial component of the 3D modelling pipeline. The recent highlighted

progress in depth sensor quality and measurement rates now afford rich, large depth data sets that present new challenges in terms of how best to reason about, utilise and take advantage of such profuse point data resources for the registration task. Addressing the related questions that such challenges pose form the body of work in this thesis. In this work we develop novel methodology for the registration of large collections of depth measurements *i.e. sets of sets of* points: $\{\{(x_{i,j}, y_{i,j}, z_{i,j})\}_i\}$, typically acquired from varying viewpoints i of physical object surfaces. The wide ranging applicability of point set registration has lead to a large body of work on the topic (see 1.3 and Chapter 2). Yet, with large point data sets now routinely generated by the outlined depth capture methods, there is renewed motivation to develop novel applications capable of utilising these data and exploring the resulting advantages that doing so brings. This reasoning underlines and influences the direction of work undertaken in the thesis.

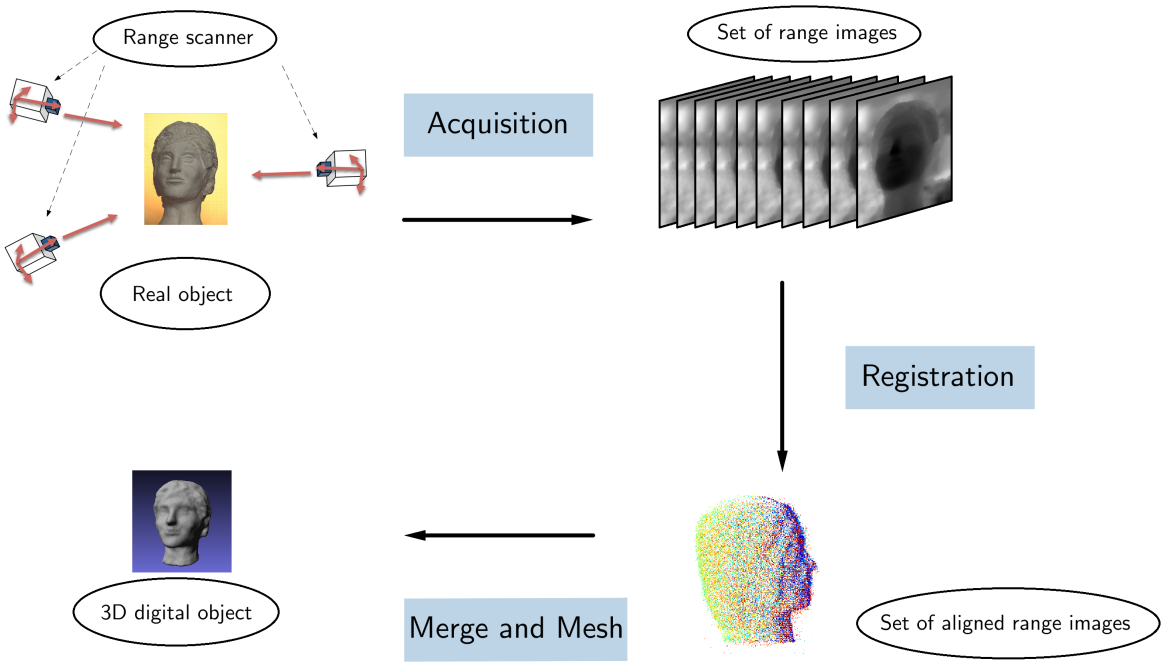


Figure 1: The model reconstruction pipeline.

1.3 3D POINT SET REGISTRATION

The crucial stage of the *modelling* process, considered in this work, is the global *registration* (alignment) of all sensor viewpoints in a common reference frame. The spatial

transformations that relate point sets / range images are in general unknown and finding them is necessary to register all overlapping point sets into a global coordinate system. This process is commonly known as range image or point set *registration* [226] and is one of the central problems tackled in this thesis. This problem is typically tackled either by identifying correspondences between adjacent viewpoints or by minimising cost functions that model the viewpoint alignment quality. Once range images have been aligned (*registered*) they can be merged and integrated into a single 3D model (*e.g.* a mesh or cloud of points) that can be utilised by *e.g.* the aforementioned example application areas. Application requirements are normally focussed on accuracy, robustness, automatism and computational speed. In this thesis, after exploring registration accuracy and robustness, we go on to investigate methodology that also enables the latter requirement regarding computational feasibility.

We summarise the 3D modelling process in Figure 1. One of the most critical tasks for the automation of a 3D modelling pipeline is automating the outlined registration step. A large body of previous work exists in particular for the *two view* and relatively small *multi-view* set cases of registration. In these forms, the registration step of the pipeline has attracted a large amount of interest in recent decades. Aligning overlapping range images using geometry is the most popular approach to 3D registration and extensive study of progress to date is surveyed in Chapter 2. When using geometry to guide the registration process discriminative feature descriptors are often used to identify *sparse* key-point correspondences that allow estimation of spatial transformations between viewpoints. Popular descriptors include the position of the key-point and the normal or curvature on the surface at this point. Alternatively a *dense* 3D registration approach, typically able to afford finer adjustment towards the desired solution, involves the minimisation of a cost function that models the quality of alignment. Proposed cost functions typically represent the distance between two aligned range images or the geometrical distribution of points. Impressive progress and seminal registration techniques [21, 163, 14, 137] have been proposed and many studies surveying 3D point registration exist (see Chapter 2 sections 2.1 - 2.2) however with the advent of recent commodity depth sensing hardware, able to afford hundreds or thousands of viewpoints quickly, new methodology able to cope with and harness this abundance of readily available data is clearly required. While depth data can now easily be collected in massive volumes, in raw form it does not provide a semantic understanding of the environments captured.

Such data does however provide an opportunity to discover and understand variability in shapes, both in terms of their geometry and their arrangements. Registering large view-sets of multiple range images remains challenging and generates theoretical and computational questions that remain open such as how best to register hundreds or thousands of viewpoints simultaneously as part of a modelling pipeline. In this thesis we explore methodology capable of handling this new excess of available depth data and explore the model reconstruction accuracy benefits that a wealth of well registered viewpoint data can afford.

1.3.1 *Point registration: problem classes*

The 3D point set registration component of the modelling pipeline essentially involves solving the problem of bringing together two or more shapes that represent parts of the same object. As outlined, registration constitutes a critical and necessary stage in 3D modelling (see section 1.1) however the goal of finding an optimal registration between several instances of the same object (or distinct but similar objects) and bringing the 3D data into a common global frame of reference is commonly utilised in additional pipelines and problem classes. Problem classes can be arranged into the following categories and the interested reader may consult [200, 252] for further discussion of categories that successful registration techniques may be applied to.

Model reconstruction. Model reconstruction is the main problem class concerning point set registration considered in this thesis. As discussed the aim of model reconstruction involves creating a complete object model from partial 3D views obtained using a depth sensing system. Due to sensors not being omnipresent it is rare that a single depth view is able to capture complete object structure, due to (self-)occlusions and sensor field of view. By capturing an object from multiple points of view and making use of successful viewpoint registration, one is able to produce an alignment between the partial overlapping views resulting in a complete object model, also known as a mosaic (see Figure 1). When treating multiple viewpoints, often registration is first applied between pairs of views [21, 221]. The entire model can then typically be reconstructed using multi-view registration refinement [130, 221]. It is the proposal of novel multi-view registration methodology (Chapter 3) and the application of these contributions to extremely large

view sets (Chapter 5) that comprise the core contribution of this thesis. As stated in section 1.1, the model reconstruction process can be applied to many application areas.

Multimodal registration. If several views of the same object or scene are acquired from different modalities (types of acquisition system) then the alignment task becomes multi-modal registration. Registered information from different modalities can be combined for comparison purposes or for creating multi-modal object models. This registration problem instance is typical in medical imaging where *e.g.* MRI and CT sensor data or MRI and PET scans [167, 233] can be co-registered. See [200] for further discussion on medical image registration. The large scale computational methodology proposed in this thesis (Chapter 4) has additionally been utilised in a medical image registration setting [212].

Model fitting. By finding optimal transforms between partial 3D depth data, acquired from a physical object, and a known model of the object (*e.g.* a CAD model) model fitting can be performed. The common applications of performing this task include robotic object grasping [107, 196] and (model-based) object tracking [67]. Model fitting has historically been applied to rigid bodies with recent work extending this to deformable objects [48].

Object recognition. If a database of known 3D models is possessed, one can perform registration between each model and a partial 3D depth sensor view (query) as a means for finding the best available matches. This problem is often regarded as more challenging than model fitting [200] as there is a decision point regarding which database model (if any) provides a correct match. This is *recognition-by-fitting* [264] and such techniques have been applied to both 3D face recognition [31, 36, 229] and object retrieval [101, 254]. The registration task component for this problem class often becomes more challenging in cluttered environments, containing many objects [139, 182, 13].

1.3.2 *Point set registration: problem formulation*

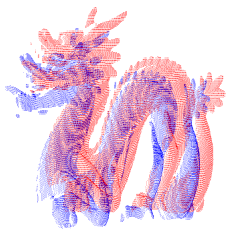
The point set registration problem can be formalised as follows; given a pair of viewpoints $\mathbb{D}$ and $\mathbb{M}$, representing two sets of points (partial 3D viewpoints of the same

object), the problem of registration involves finding parameters θ of the transformation function $T(\theta, \mathbb{D})$ that brings $\mathbb{D}$ into the best spatial alignment with $\mathbb{M}$. By convention, in the two view case, we name $\mathbb{D}$ and $\mathbb{M}$ the *data* and the *model* and point sets are typically represented as point clouds or triangulated meshes [43]. In this thesis we utilise point cloud data representations from a variety of data sources and depth sensors.

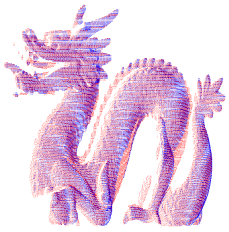
In the two-view case, the moving view $\mathbb{D}$ (the *data-view*) and the fixed view $\mathbb{M}$ (the *model-view*) can be aligned by solving the registration problem that estimates the parameters θ^* of the transformation function T that satisfy:

$$\theta^* = \arg\min_{\theta} E(T(\theta, \mathbb{D}), \mathbb{M})$$

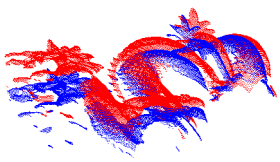
where E is an *error function* that quantifies the registration error. Figure 2 illustrates a typical input and result of the elemental two-view registration process, searching for an optimal rigid spatial transform between two viewpoints of the same rigid body object.



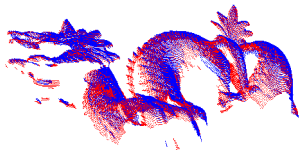
(a) *Dragon* dataset pre-registration (horizontal view)



(b) *Dragon* dataset post-registration (horizontal view)



(c) *Dragon* dataset pre-registration (isometric view)



(d) *Dragon* dataset post-registration (isometric view)

Figure 2: An example of the Iterative Closest Point registration algorithm applied to two point clouds. Starting pose (Figures 2a, 2c) and views post-registration (Figures 2b, 2d). Dragon model data provided by the Stanford 3D Scanning Repository [258].

The moving data-view (depicted in red) and model-view (blue) point sets provide measurements for different portions of the surface with non-zero overlap (Figure 2a, 2c).

The optimal transformation parameters found[1] are applied in the function $T(\theta, \mathbb{D})$ and the resulting registered views are rendered for comparison (Figure 2b, 2d).

Extending this formulation to generalise to the case of multiple views; consider a transform G_i taking the data in a common global frame of reference into the coordinate frame of view i. Similarly, let T_{ij} be a transformation resulting from an optimal pairwise registration transforming the points of view j into the coordinate frame of view i.

If perfect (noise-free) pairwise registrations can be found then $T_{ij} * G_j$ and G_i share an identical frame of reference for all view pairs i and j. When formulating multi-view registration in this manner, the task commonly becomes one of finding a set of rigid motions from each view to a common global frame of reference yet also satisfying the locally optimal, potentially conflicting transform constraints obtained from the *pairwise* registration of i, j view-pair permutations.

This problem is often formulated as an error minimisation. By defining an error Err to minimise (*e.g.* Euclidean distance between point locations) and composing local and global components such that optimal pairwise transforms found are applied in the global frame of reference, we can seek to minimise the function:

$$Err = \sum_{i} \sum_{j \in \text{Neighb}(i)} \sum_{\mathbf{p} \in \mathcal{P}} \| T_{ij} * G_j \mathbf{p} - G_i \mathbf{p} \|^2 \tag{1}$$

Here Neighb(i) is typically the set of all views that exhibit sufficient overlap with view i and this formalisation of the multi-view problem is similar to (for example) the approach taken by Pulli [210], where the error function E is defined to be the Euclidean distance between point locations (see Eq 1). In this case, $\mathbf{p} \in \mathcal{P}$ represent a set of (arbitrarily chosen) spatial points $\mathcal{P}$ such that $\|\mathcal{P}\| \geq 3$ to define registration unambiguously. More recently [263] formulate the problem in a similar manner yet adopt a different measure of error that derives from the fact that any rigid transformation is in fact a screw motion; thus defining a *screw distance*.

This generalised registration quality formulation is similar in spirit to a standard ICP error [21], but differs in that (1) here error is measured between identical points in potentially differing spatial positions, not distinct points in correspondence (2) $\mathbf{p} \in \mathcal{P}$ need not be a set of sampled points from any particular view i.

1 In this example an instance of the classical ICP algorithm [21] is used for demonstration.

While this formulation (Eq 1) is adequate for small instances of the multi-view problem, studying extremely large sets of viewpoints introduces compounding problems involving prohibitive combinatorial view pairing considerations. Even in instances where valid and meaningful view combinations are reliably known *a priori*, the task of finding pairwise motion parameters for each pairing may become excessively time-consuming, costly and *pairwise* optimal may not be *globally* optimal. This observation motivates the alternative objective formulations presented in the current work, solving the multi-view registration problem to provide motion parameters that bring each viewpoint into a consistent global frame of reference, without treating pairwise registration explicitly. The resulting formulations retain the ability to solve the multi-view registration problem yet remain applicable to large view-set problem instances (Chapter 3 provides our exposition).

1.3.3 *Point set registration: problem extensions*

The point set registration problem, formulated above, is a heavily studied problem in computer vision. However, several extensions of the generic task remain challenging and related open questions continue to emerge. The registration problem becomes more difficult when (1) more than two views must be brought into the same frame of reference (*multi-view* registration - section 1.3.2), (2) registration must be performed in *cluttered scenes* and (3) registration includes *deformable objects*. Figure 3 (adapted from [200]) illustrates a taxonomy of these advanced registration problem sub-classes and current challenges relating to each.

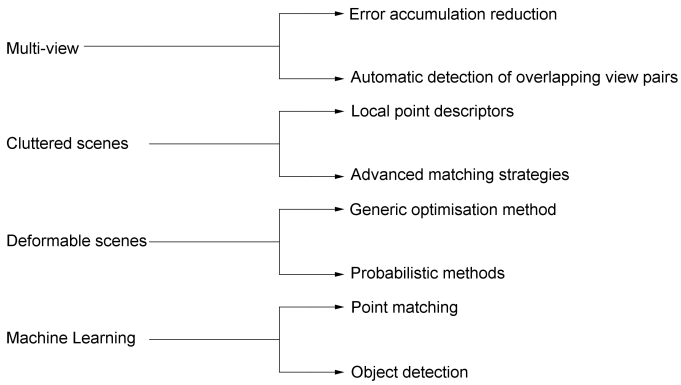
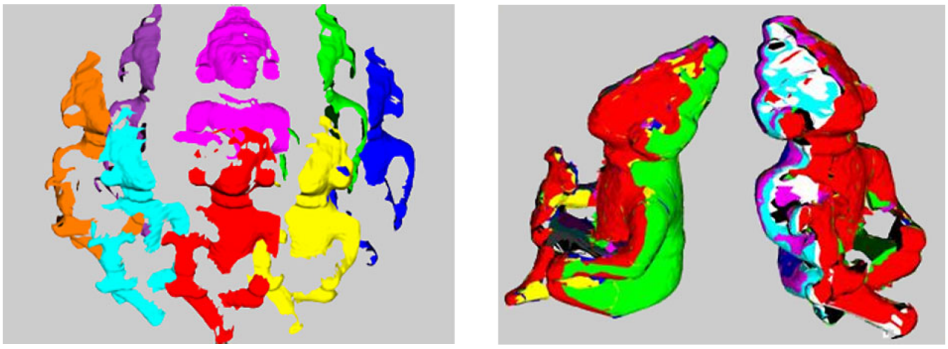


Figure 3: A taxonomy of advanced registration tasks and the related challenges posed. Figure adapted from [200].

This thesis will predominantly focus on the advanced point set registration problem instance concerning *multi-view* registration. Previous work in this area of the outlined taxonomy is treated in Chapter 2 accordingly. When the number of viewpoints to be registered is greater than two, the set of views must all be transformed into a global frame of reference by applying multi-view registration techniques. The view registration process attempts to find optimal spatial transforms aligning all viewpoints into a global frame of reference. Figure 4 provides a schematic example of where such a technique fits into the model reconstruction pipeline.



(a) (b)

Figure 4: A model reconstruction example. Incomplete depth information relating to an object-of-interest is captured from varying viewpoints using multiple sensor poses (see 4a). Figure 4b shows the merged views in a common coordinate system post-registration (see text for detail). Images adapted from examples generated by Alessandro Negrente [200]. 3D data and object model attributed to [102]. Best viewed in colour.

Multi-view registration has often historically been attempted as an additional step after pairwise registration has been performed between combinations of pairs of viewpoints. Oft-cited issues with such multi-view registration strategies are *error accumulation*, *error propagation* and the level of *automation* achieved in the process. Alternative forms of preparatory coarse initialisation such as manual alignment have also been utilised, see Chapter 2. If viewpoints are registered in a linear or chained pair-wise fashion, local pair-wise alignment error may propagate between viewpoint pairs and grow. Additionally if a full model must be reconstructed from a large number of scans, the view order may not be available and therefore may have to be manually specified for pair-wise combinations.

This thesis develops novel methodology that attempts to address these multi-view specific problems by treating viewpoint alignment as a global optimisation problem rather than a chain of sequential local alignments. The tactic of performing registration *simultaneously* among all viewpoints has been attempted and utilised previously (see Chapter 2 section 2.2) as the noted benefits of avoiding sequential registration error accumulation and propagation are desirable. However such global optimisation is often known to be computationally expensive and we find that registration methods often scale with variables such as the number of viewpoints to be aligned and the density of point sets utilised. Such computational cost considerations are in conflict with our goal of exploring multi-view registration in cases where hundreds or thousands of viewpoint measurements may be available. It is this combination of the favourable properties of global optimisation and large scale problem instances that primarily motivate the methodology and ideas developed and investigated in this thesis.

1.4 SUMMARY AND OUTLINE

In this chapter we introduce the standard 3D modelling pipeline and defined the viewpoint alignment and point set registration components of such a framework. Point set registration is a well studied problem yet specific problem instances leave issues that remain to be solved. Specifically, advances in sensor hardware and progress in depth acquisition procedures provide new challenges and tests for multi-view registration methods aiming to firstly accommodate hundreds or thousands of viewpoints, finding globally optimal alignments and secondly utilise the benefits that such large data sets are able to afford the multitude of noted 3D modelling application areas. These observations motivate the work proposed in this thesis. In particular prominent problems remain to be addressed when modern depth scanners, capable of providing extremely large sets of measurements, are made use of. Real-time depth scanners, for example, provide an explosion of the number of depth data measurements that can be acquired in relatively short time periods. If viewpoint registration components are required in scenarios utilising such sensors then exhaustive search should likely be avoided and more effective point matching strategies and error evaluation functions should be exploited. Both (1) quantitative registration error evaluation functions, capable of capturing the nature and intrinsic properties of large scale multi-view alignment problem instances and (2)

their implementation utilising robust and scalable frameworks are needed to handle the noted explosion of available depth data to provide desirable results that can *e.g.* contribute to high quality modelling pipelines in timely fashion.

In this thesis we firstly define a density estimation technique that we propose meets the former of these requirements. We show how kernel density estimation can be utilised to construct an estimate of a sampled surface, based on measured spatial data and apply this technique to data samples from a variety of depth sensors. We proceed to illustrate how this tool can be utilised to enable the extraction of useful information from sampled data (*e.g.* for viewpoint registration). Registration errors can typically be attributed to a lack of high-level, cross-viewpoint understanding about the entities represented in the scene. We claim that by designing models that incorporate a global level of understanding about object and scene shape and combining this with the ability to accommodate many sensor measurements we apply our techniques to challenging datasets of increasing size and complexity whilst leveraging the confidence that many sensor samples afford. The latter claim, that such techniques should be implemented in robust and scalable frameworks enabling the accommodation of data sets on an order of magnitude afforded by modern depth sensors, is then investigated by the proposal of a distributed and parallelisable task farming framework. Finally our multi-view registration theory and our contribution of a scalable and parallelisable framework are combined to explore the feasibility and benefits of performing multi-view registration with extremely large view sets under such an implementation in practice.

1.5 THESIS CLAIM

An important premise of this thesis is that large view-sets and an abundance of depth data are beneficial in terms of model completeness and accuracy and can be utilised advantageously when tasked with modelling object surfaces and shape. By implementing a methodology capable of building models of object surfaces from large sets of partial viewpoints, we aim to exploit the hypothesis that discrepancies, due to *e.g.* measurement error or sensor noise, between measured surface data and estimated models will tend to zero if enough samples are present. This premise can be captured in the following claim:

By registering partial views simultaneously to a robust surface estimate, it is possible to improve registration accuracy over sequential approaches by distributing errors evenly between overlapping viewpoints. Object surfaces can be robustly estimated from coarsely misaligned partial views using density estimation techniques and such estimates can be utilised to reliably guide simultaneous point cloud registration. This approach exhibits an inherent ability to handle data from many viewpoints simultaneously and improves registration and reconstruction accuracy over existing techniques by exhibiting robustness to initial coarse misalignment of view-sets.

This thesis defends this claim by designing techniques that aim to accommodate and leverage an abundance of viewpoint information for the registration task. By utilising large viewpoint datasets and quantitatively evaluating registration results produced by the proposed methodology we assess the claimed benefits of accounting for increasingly large and complex viewpoint data.

A main pragmatic goal of the thesis is to facilitate accurate simultaneous registration of large sets of point clouds in a global coordinate frame. By employing data-driven density estimates we aim to estimate object shapes, contributed to by large quantities of depth sensor viewpoints. Both the computational speed of density estimation and quality of resulting models typically depend on the number of data samples available. Intrinsic properties of non-parametric density estimation dictate that estimation quality improves as the number of available samples increases however estimation often also becomes more expensive. This non-parametric estimation property essentially dictates that the cost of building models will increase as the number of available samples to be utilised increases. We mitigate the computational cost of the proposed approach by additionally introducing a parallelisable framework capable of distributing the workload and overall improving methodology feasibility. View sets experimented with in the latter chapters of this thesis are on an order of magnitude that is infeasibly large for traditional serial and sequential point cloud registration methods to accommodate. The potential available benefits of building models of objects and scenes from data sources containing viewpoint counts that are $1-2$ orders of magnitude greater than traditionally available can thus be explored.

1.6 OUTLINE

In summary the remainder of the thesis consists of five chapters and is structured as follows:

- In Chapter 2 the state of the art of point set registration is summarised and additionally an overview of contemporary distributed computing and task farming techniques is provided. Areas of interest are highlighted and open challenges identified.

- Chapter 3 presents a novel *simultaneous* multi-view registration technique that makes use of kernel density estimation theory. Details of how the proposed approach enables the registration of a set of point clouds are presented. The strategy does not make use of direct point pair correspondences or require any view ordering information. The work presented in this chapter is published as follows:

  Simultaneous registration of multi-view range images with adaptive kernel density estimation. S. McDonagh and R. B. Fisher. *14th IMA Conference on Mathematics of Surfaces*. pp 31–62, Birmingham, 2013 [176].

- In Chapter 4, we introduce a framework that we call Semi-Synchronised Task Farming (SSTF). The proposed framework provides a principled method for implementing computationally expensive problems in a distributed fashion across heterogeneous compute clusters. By formulating compute problems as a collection of parallelisable subtasks and enabling a level of communication between subtasks we present a framework and novel computation model able to produce predictable speed-up improvements to computationally expensive yet non-trivial work. The framework presented in this chapter was introduced in:

  Applying semi-synchronised task farming to large-scale computer vision problems. S. McDonagh, C. Beyan, P. X. Huang and R. B. Fisher. *International Journal of High Performance Computing Applications*, 2014 [178].

  Additionally the computational framework has recently been utilised to perform 2D intensity image registration:

Laminar and Dorsoventral Molecular Organization of the Medial Entorhinal Cortex Revealed by Large-scale Anatomical Analysis of Gene Expression. H. L. Ramsden, G. Sürmeli, S. McDonagh and M. F. Nolan. *PLOS Computational Biology*, 2015 [212].

- Chapter 5 presents an implementation of our proposed multi-view registration strategy under our distributed SSTF framework. This facilitates simultaneous registration of extremely large sets of point clouds in feasible time frames. With this system we explore the available benefits of performing feasible, large-scale view registration and building object models and scenes from data sources containing view counts that are $1-2$ orders of magnitude greater than traditionally available. The work therefore explores improving view registration and model reconstruction quality when applying our point set registration framework to extremely large sets of object viewpoints that potentially contain multiple and redundant depth samples of physical points captured from varying views. This is made feasible through the use of our distributed framework introduced in the preceding chapter.

- Finally, Chapter 6 concludes and summarises the work carried out in this thesis and provides discussion on potential future direction for large-scale point set registration in relation to the conclusions attained in this work.

Part II

BACKGROUND

2

LITERATURE REVIEW

The topic of image registration has produced a large body of work and remains the subject of extensive effort. Interest stems from the challenging nature of the associated fundamental problems (*e.g.* point correspondence definition, transformation model selection) and the importance of solving these challenges for various applications. Initial efforts focused on registering 2D images as these originally constituted the most commonly available data. The related early survey paper of [144] mainly covers work based on image correlation. Several exhaustive reviews of general-purpose image registration methods have additionally been produced since *e.g.* [37, 277, 290]. Registration techniques applied particularly in medical imaging are summarised in [268, 156, 167, 124]. Restricting scope to surface based registration, medical imaging applications are surveyed by Audette et al. [11] while volume-based registration is reviewed in [74]. Additionally registration methods applied in remote sensing settings have been described and evaluated in [95] and [186].

With the advent of active 3D sensors (*e.g.* structured light sensors, laser range finders) and the progress of passive stereo vision, registration of 3D image data has also grown into a substantial topic in the computer vision literature and a review of the evolution of range image registration methods can be found in [226]. Additionally the recent rapid advances of commodity depth sensing hardware (*e.g.* Kinect [183]) afford abundant, widely available streams of high frame rate, low-cost depth data. A main contrasting characteristic of depth image data, relative to 2D images, is that working with depth images (or other range data) allows 3D scene and target geometry information to become directly available. This has presented new possibilities and new challenges for the topic of image registration.

In this chapter we provide a survey of the state of the art in range image and point cloud registration techniques. A registration technique's efficacy is determined by its accuracy and the computational complexity of its component algorithms. We review several classes of techniques that have been utilised for the task of point cloud registration in the literature. The objective here is to provide a high-level overview, we leave more detailed comparisons of methods, related to the work presented in this thesis, for consideration in the chapters that follow. We begin by considering two-view point set rigid and non-rigid registration approaches in section 2.1 and review techniques that concern multi-view registration in section 2.2. We discuss methods attempting to solve large view-set problem instances and methodology to evaluate registration quality in section 2.3 and touch on related distributed computation issues, relevant to this thesis, in section 2.5. Finally we conclude with a summary in section 2.6.

2.1 TWO-VIEW POINT SET REGISTRATION

Many algorithms have been presented for both rigid and non-rigid point set registration. The typical goal of these algorithms is to recover correspondences between points, a transformation[1] to align the point sets, or both. Commonly these algorithms involve an iterative dual-step update, alternating between point correspondence search and transformation estimation. When considering only two scans or viewpoints (*pairwise* registration) the problem can be considered well studied in the computer vision literature. Early work on pairwise point set registration and scan alignment was performed by Faugeras and Herbert [89], Horn [128] and Arun et al. [9]. In each of these examples the authors obtained closed-form expressions for a single rigid transformation that minimised the least squares error between the two point sets.

The influential Iterative Closest Point (ICP) algorithm proposed by Besl and McKay [21] is the most popular method for rigid point set registration due to its simplicity and low computational complexity. ICP iteratively assigns point correspondences based on the closest distance criterion (considering minimum Euclidean point distance pairs between sets) and then finds the least-squares rigid transformation that best aligns the two point sets using the found correspondences. Figure 5 depicts the basic algorithm. The literature contains work exploring additional metrics for spatial point set registra-

1 transforms can be sub-categorised as rigid, non-rigid and pointwise deformations

tion (*e.g.* photometric constraints [232, 259]) however distance metrics remain popular. When utilised for point cloud registration, this approach typically starts from a coarsely aligned seed pose and iteratively revises a transformation (typically composed of rotation and translation in the rigid case) to minimise the distance between pairs of neighbouring points in the two point clouds. Subsequent proposed variants of ICP affect all attributes of the algorithm including the selection and matching of point correspondences and the minimisation strategy (see *e.g.* [54, 94]). Variants have modified the point pair matching strategy by *e.g.* rejecting conflicting point pairs or weighting correspondences with similarity measures, and varying the minimisation metric (*e.g.* point to (tangent)-plane distances [54]). Recent work carries out robust correspondence search utilising a novel graph matching strategy in combination with graduated nonconvexity and concavity [281]. Additional work has involved altering the measures of alignment error and employing data structures (*e.g.* k-d trees) to facilitate fast point pair search. The family of techniques broadly based on local iterative decisions thus are generally susceptible to local minima (for example, when poor initial coarse view alignment is provided). ICP and variants typically therefore require that the initial position of point sets be adequately close.

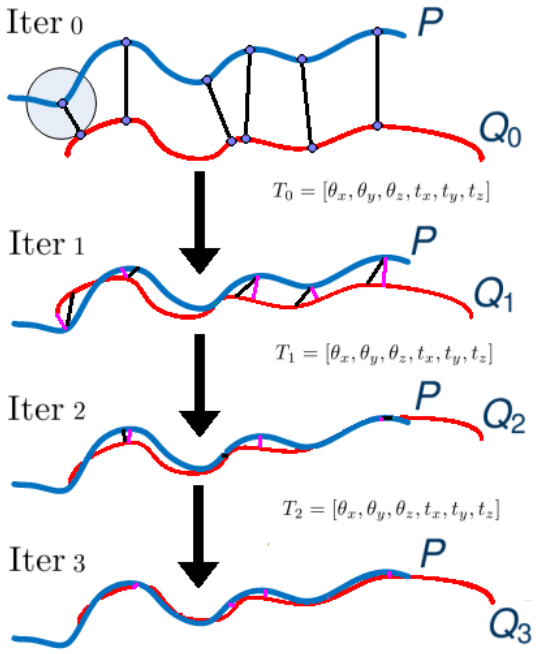


Figure 5: The fundamental Iterative Closest Point algorithm [21] for registering two point sets. Point pair correspondences and optimal spatial transforms are alternatively found and this process iterates to convergence.

Probabilistic variants have been developed (*e.g.* [213, 23]) in order to produce an alternative to assigning hard point correspondences between point sets. These methods commonly use soft assignment of correspondences that establish a global correspondence between all combinations of points according to some probability. This strategy effectively generalises the binary assignment of correspondences found in the original ICP. Among these methods are the Robust Point Matching (RPM) algorithm introduced by Gold et al. [106], and its later variants [213], [58]. In [56] it is shown that alternating soft assignment of correspondences and transformation refinement in a RPM setting is equivalent to the Expectation Maximisation (EM) algorithm, where one point set is treated as a set of GMM centroids and the other point set is treated as data points. Several further rigid registration point set methods (*e.g.* [142],[276],[64],[23],[111],[164] and [180]) explicitly formulate point set registration in a Maximum Likelihood (ML) estimation framework to fit GMM centroids to data points. These methods re-parameterise GMM centroids by a set of rigid transformation parameters (representing translation and rotation). The EM algorithm, used to optimize the likelihood function, consists of two steps; an E-step to compute the probabilities and an M-step to update the transformation. Common to such probabilistic methods is the inclusion of an extra distribution term to account for outliers ([213, 276]) and annealing to avoid local minima (poor registrations). Such probabilistic methods have been shown to offer improved performance over the original ICP algorithm, especially in the presence of noise and outliers. A comprehensive review of ICP variants is found in [221]. Recently probabilistic generative model based approaches have also been extended to construct algorithms capable of jointly registering multiple point sets [87] and we survey the array of multi-view approaches in the following section.

Addressing matching speed improvement, Blais and Levine [25] minimise a Euclidean distance cost function calculated on sets of control points and utilise simulated annealing to perform a projection-based ICP. Silva et al. [238] adopt a similar approach but use genetic algorithms with a surface inter-penetration measure. In [16], a randomised ICP over a multi z-buffer structure is proposed. The structure is capable of representing overlapping portions of the viewpoints and accelerates operations on them. An improved force-based optimisation method is also proposed by Eggert et al. [83], [84]. Alternative representations of rigid rototranslations have also been explored. Quaternion representation of rototranslation transforms [128] are made use of in several global

registration works including [15], [235], [236] and [263]. In [15] it is demonstrated that the optimal translation can be decoupled and solved independently from the optimal rotation. The approach is based on iteratively finding rotation solutions by moving one view at a time while keeping the other viewpoints fixed in space. A similar decoupling is exploited in the work of Sharp et al. [235, 236] where optimisation over the graph of neighbouring viewpoints in a quaternion space is done and then closed form solutions are obtained using the cycles of a graph decomposition. One of the advantages of this method is that it does not require the computation of point correspondences and can be combined with any pairwise alignment algorithm to generate an estimation of the relative motion between each pair of views.

Previous work of key foundational importance, highly relevant to the correspondence methodology proposed in this thesis is found in [265]. This work provided the original contribution of extending image correlation techniques to the point set registration problem, using kernel correlation (KC). A kernel correlation affinity measure is defined as a function of the point set entropy between two point sets to be registered. This registration method has the advantage of an intuitive interpretation and convergence properties, with the proposed algorithm comparing favourably to both ICP and EM-ICP based methods. Maximum KC between only two points corresponds to the minimum Euclidean distance between them however, when dealing with multiple point distances, it's not immediately obvious what is being optimised. Interestingly the authors highlight the observation that maximising their KC (essentially a product of kernels) turns out to equate to minimising the Euclidean distance, but in the sense of an M-estimator (in the case of the Gaussian kernel that they make use of). This is related to the work that will be proposed in this thesis in that we make use of similar robust kernel methodology to define (novel) objective functions for the purpose of performing the (multi-view) registration task while still taking advantage of all the useful properties of a kernel based optimisation highlighted in [265].

In summary, the registration of two-view point set data is a well studied problem. The standard ICP methodology [21] involving iterative search for point correspondences followed by defining optimal transforms can be considered the most popular strategy. This iterative approach has been applied to many problem instances and is relatively straightforward to implement making it a popular choice. The well understood shortcomings of the original version make it somewhat limited by current standards however

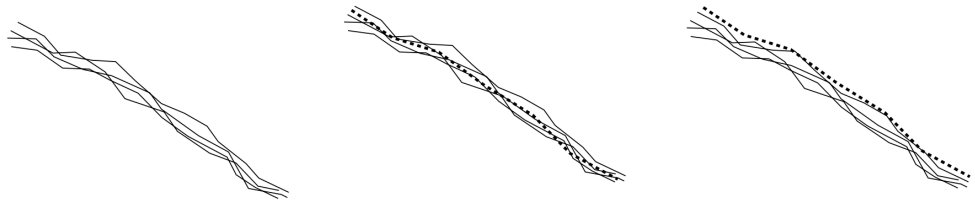
several strong variants and work inspired by ICP (*e.g.* [55, 188]) allow the strategy to cope with a wide range of scenarios. The interested reader can examine comprehensive surveys of the wide array of two-view registration techniques in a number of reviews [37, 221, 181, 226, 252, 204]. When multiple views are to be considered, additional complications arise which we discuss in the following section.

2.2 MULTI-VIEW POINT SET REGISTRATION

When considering multiple views, view poses must be transformed into a global reference frame using a multi-view registration technique. Common issues with multi-view registration involve automation of the process, error accumulation, error propagation and loop closure issues. Reducing error accumulation when view sequence ordering is available allows for registration to be performed in a pairwise fashion between consecutive views. In general, even if all the pairs are visually well registered misalignment typically appears when the full model is reconstructed due to the accumulation and propagation of sequential, incremental registration error. Multi-view registration techniques often introduce additional constraints that reduce the global error. This strategy is commonly embodied by solving simultaneously for a global registration, exploiting the interdependences between all views at the same time. In this sense multi-view registration generalises the case of two-view registration and often poses a more challenging problem. Typically ten or more views are utilised to reconstruct a complete object model, where each viewpoint overlaps a number of neighbouring scans. Creating accurate 3D models of real objects is a primary goal of several application domains such as industrial reverse engineering [269], visual inspection [34], cultural heritage preservation [271], robot localisation and navigation [30] and biological and medical imaging [220].

Two popular approaches to multi-view registration are sequential (local) registration and simultaneous (global) registration. Early sequential techniques such as those proposed by Chen and Medioni [54], Masuda and Yokoya [173] simply align pairs of overlapping views in turn. The points of each aligned view are then merged into a meta-view until each view has been aligned by registering the next scan directly with all merged data from the previously processed and registered views. A potential problem with this meta-view approach was highlighted by Pulli [210]; if several scans are

added to a meta-view, a shell of finite thickness will likely be created. When yet another scan is registered with the meta-view, ideally the new view would move into a central position among the previously aligned scans however by minimising a standard point-to-point distance metric (or similar) the new scan is likely to stick to the outer or inner shell of the meta-view (see Figure 6). Due to such potential drawbacks, final solutions are in general suboptimal if no global optimisation of view positions is performed. When many views are registered in a sequential fashion by designating a base scan or another previously registered viewpoint as model data (*e.g.* using pairwise ICP) the resulting registrations may have low quality when combined.



(a) Piece of the meta-view (b) Ideal registration for a new scan (c) Likely new registration

Figure 6: A common problem with the meta-view approach. Figure adapted from [210].

Sequential techniques additionally require that view sequence order is known or manually specified in advance, using prior knowledge to guide which pairs of scans alignment should be attempted between. This information is now often available (*e.g.* using modern video-rate active depth cameras) however it remains a necessity that each pair of registration candidate scans has substantial overlap to result in successful registration. When a coarse alignment initialisation is unavailable, an alternative sequential strategy involves exhaustively attempting to register each pairwise combination of viewpoints and implementing a method for determining registration success between scan pairs (*e.g.* visual assessment or quantitative resulting overlap measurement). Exhaustive pairwise matching strategies quickly become infeasible as the number of scans (viewpoints) grow large. Such approaches have, however, proved popular due to method simplicity and the relatively cheap and fast solutions that are offered despite the noted issues involving pairwise error accumulation and propagation that may lead to globally suboptimal results.

Popular pairwise schemes [21, 54] are still often utilised as components in multi-view approaches. Multi-view registration schemes of this type were considered by numerous researchers [19, 84, 15, 238, 236, 279]. Pairwise registration error accumulation and propagation is addressed by Bergevin et al. [19] where points in each view are matched with all overlapping views and a rigid transform that registers the active scan is computed using the matching points from all overlapping views. By organising pairs of views in a network structure they enable simultaneous and iterative alignment error minimisation. By making use of all overlapping views this approach attempts to diffuse errors among all viewpoints as the process is iterated to convergence. Converging to a steady-state using this approach may be slow and computationally expensive. A similar iterative approach that computes the "mean rigid shape" of multiple point sets was proposed by [201] but point correspondences had to be specified manually as a point matching algorithm was not included. An early numerical solution is proposed by [249] where the registration problem is mapped onto a physically inspired model where a minimum of potential energy is found with an iterative numerical method based on gradient descent yet slow convergence may occur (particularly with cases of near degenerate point sets). Further early multi-view work by Eggert et al. [83] constrain the point pairings such that points of each scan match with exactly one other point and then minimise the total distance between paired points. The transformation update is then solved for by simulating a spring model. The registration work presented in this thesis is similar to the work of Eggert et al. in that we minimise an energy system representing scan positions but we do not constrain points to an individual point-to-point match (see chapter 3 for further detail). A review of comparable early multi-view registration methods [201, 249, 15, 16] was carried out by Cunnington and Stoddart [65] and further recent works, that this comparison pre-dates, are outlined in more recent comprehensive surveys [226, 200, 252].

Global multi-view techniques attempt to mitigate the discussed sequential registration problems by taking all scans into consideration at once, thus attempting to spread registration error evenly between all overlapping views. Extensions of the ICP algorithm in particular have been proposed for simultaneous registration of multiple range images, however handling multiple range images simultaneously will often dramatically increase computational time for an ICP style approach. As shown by [84] it takes $\mathcal{O}(r^2 N log(N))$ operations to find all point correspondences across pairs of r point sets with N points

each. Computing potential correspondences is generally the most time consuming step and such methods therefore become impractical as the number of range images become large. One of the early fully simultaneous registration works was proposed by Pulli [210] where following pairwise scan alignments, each pair of registered scans are used as constraints in a multi-view step. As introduced in the previous chapter, the goal of diffusing the pairwise errors is achieved by first aligning the scans in a pairwise fashion and then utilising these pairwise alignments as constraints in a simultaneous step. The aim is to evenly distribute the pairwise registration error, but the method itself is still based on initial pairwise alignments. Pulli's algorithm remains a method of choice for many multi-view point set registration applications (the Scanalyze software [227] provides a popular implementation). Formally Pulli attempts to keep the distortion $\mathcal{D}(U)$ of the points from a set U within a given tolerance ϵ where we define $\mathcal{D}(U)$ as:

$$\mathcal{D}(U) = \sum_{u \in U} \sum_{(i,j) \in \mathcal{V}} P_i(u) - T_{i,j}(P_j(u))^2$$

In this formulation $P_i(u)$ is a transformation that transforms a point u into the coordinate system of view i while $T_{i,j}$ is the transform that maps the coordinate frame j into the coordinate frame i (as found by the pairwise registration between the two frames) and $\mathcal{V}$ is the set of neighbouring view pairs for which pairwise registration is carried out. The set of points U on which to perform this greedy approach must be specified and Pulli suggests that these points can be sampled uniformly from the overlapping areas of the scan views. Since only the space of transformations is explored in this approach, memory usage is small as there is no need to retain all of the points from all views in memory at once. This allows for global registration on data sets that are too large to keep directly in memory. There is, however, no guarantee that optimal solutions are found. Williams and Bennamoun [279] took a similar approach attempting to minimise a similar distortion on a set of sampled points, computing the minimisation using an iterative approach and optimising individual transforms using singular value decomposition.

Failure modes of the method that remain include the handling of multiple closures problem (from the acquisition of complex objects) as well as when the number of viewpoints to align increases up to a point that the underlying heuristic fails to converge to the global minimum of the error function. Our experimental work in chapter 5 (and other recent work [27] explore these failure modes further.

Another early simultaneous method was introduced by Neugebauer et al.[189] where a distance metric is minimised (in a least-squares sense) between overlapping range images and a signed distance function is utilised to create an intermediate volumetric model. By treating the role of multi-view registration as projecting a point transformation onto a common frame of reference, these methods attempt to reduce the accumulated pairwise registration errors. This can be achieved by limiting the difference between the position of point instances when transformed by the differing pairwise registrations. This effectively moves each scan, relative to its neighbours, as little as possible. In a similar fashion to [189], Huber and Hebert [131, 130] more recently use a global consistency measure on a graph of pairwise matching viewpoints and look for globally connected sub-graphs on which they then solve a multi-view point-to-plane distance minimisation problem. Such graph based representations, that typically assign sensor viewpoints to nodes, are used to define overlapping viewpoints by making use of edge weighting and connectivity. Such representations can in turn help to define heuristics and algorithms to better condition the problem or to retain feasible computational cost. Alternative options in practice favour a greedy approach (*e.g.* [210]) to limit the difference between the locations of point sets as they are positioned in two frames (when transformed by relevant pairwise registration transforms).

In [131] a global optimisation process searches a graph constructed from pairwise view matches to provide a connected sub-graph containing only correct matches, using a global consistency measure to eliminate incorrect but locally consistent matches. Further approaches use *global* and *local* pre-alignment techniques to select overlapping views and compute a coarse alignment between all pairs of views. In [168] a pre-alignment is performed, first extracting global features from each view, namely extended Gaussian images. Conversely in [147], a pre-alignment is computed by comparing the signature afforded from feature points. After these have been compared the best view sequence is estimated by solving a Travelling Salesman Problem (TSP).

In [46] a method is proposed that attempts to distribute registration errors evenly between all views. It operates in the space of estimated pairwise registration matrices, however ordering of the views is required. Automating registration especially when the full model is composed of a large number of scans, the view order might not be available and therefore should be manually specified. Pottman et al. [206] develop a method based on a first order kinematical analysis that exploits local quadratic approximates

of the squared distance function associated with the surfaces to be aligned. This is investigated further in [207] where a geometric optimisation strategy is proposed and a theoretical framework is introduced in order to better interpret empirical results reported in previous work (for instance ICP-based methods exhibiting linear convergence) and in addition constrained non-linear least-squares approaches based on Newton-like descent algorithms which have been shown to lead to fast (locally quadratic) convergence. A recent scheme by Zhou et al. makes use of a clustering based approach to mitigate the effects of large accumulative registration errors and heavy scanning noise [288].

Recently Torsello et al. [263] introduced a method that extends [210], by representing view transforms as dual quaternions to project pairwise alignments onto the same reference frame and by framing the multi-view registration problem as the diffusion of rigid transformations over the graph of adjacent views (to aid error diffusion). Correspondences are allowed to vary and be updated while alternating between point correspondence choice and optimisation over the rigid transformation space (but convergence of the procedure is not discussed). This produces a similar global strategy to our framework outlined in the following chapter (see chapter 3). By alternating between the diffusion method and ICP pairwise alignment the authors apply the proposed method to real-world data where alignment performance similar to that of Pulli [210] is observed. A further simultaneous registration method for dense sets of depth images employing a convex optimisation technique for obtaining a solution via rank minimisation is introduced in [260, 259]. The work concerns depth images directly rather than point cloud data and extends previous work on simultaneous alignment of multiple 2D images. In [88] initial coarse alignment is performed by proposing a voting scheme to discover view overlap relationships and then LM-ICP [94] is extended to multiple views in order to minimise a global registration error as part of their automated registration pipeline.

Bonarrigo and Signoroni [26, 27] extend a global registration technique that aligns sets of range images using an "Optimization-on-a-Manifold" (OOM) framework previously proposed by Krishnan et al. [153, 154]. The original OOM framework proposes an unconstrained optimisation procedure that exploits translation and rotation decoupling. By making use of optimisation methods that work explicitly on the constrained manifold of rotations, $SO(3)$, they solve for the vector of all view rotations. The method guarantees a closed form transform that simultaneously registers all range images in the

noiseless case. This is under the assumption that a set of perfect correspondences are provided and the authors concede that the requirement for apriori knowledge of point correspondences from overlapping scans can be viewed as a major limitation, as this is usually not the case in practice. However, the algorithm is able to work in conjunction with methods like ICP [21], providing a general framework for multi-view registration. In the presence of noisy correspondences, this analytical solution becomes an initial estimate for any general iterative scheme. Fixing the correspondence set during a minimisation process provides one route to alleviating correspondence search cost. This has the advantage of making computation cost per iteration independent of the number of data points in each view. In the case of [154] each minimisation iteration involves finding only the inverse of 3×3 matrices, one for each view. It is this lack of correspondence updating that can be considered an enabling factor when attempting to obtain computationally affordable techniques for large view-sets (and the associated large correspondence sets of size n). Registration techniques that request exact (true) point correspondences between views as input for the task of finding closed form solutions to optimal view-set alignment are an unreasonable requirement for real scenarios and this proves a major limitation for practical applications. In real scenarios (*e.g.* where data is gathered from depth sensors and possesses only a reasonable initial alignment between views) the obtained initial point correspondences are often far from perfect.

The main novel extension that Bonarrigo and Signoroni add to the Krishnan et al. framework [154] is an improvement that allows point correspondences to be updated during the optimisation process. The method involves an error minimisation over the manifold of rotations via an iterative scheme based on Gauss-Newton optimisation that is similar to the optimisation process proposed in this work. In summary, the *point correspondence* sub-problem is generally considered to be a very important component of the registration process. The discussed methods are based on correspondence sets computed out-of-core, other work updates correspondences in an iterative manner and further work attempts to avoid using correspondences altogether. Another important aspect of the registration process involves the robustness of the found solutions. In [33] it is shown that, in the case of pairwise point set alignment, taking the intrinsic geometry of the underlying manifold into account for the purpose of registration significantly increases robustness with respect to poorly initialised poses.

Toldo et al. [262] recently proposed a global registration approach based on embedding the well-known Generalized Procrustes Analysis (GPA) mathematical theory in an ICP framework. The method iteratively minimises a cost function considering all views simultaneously and the overall strategy can be considered similar in this respect to the work presented in Chapter 3 of this thesis. Toldo et al. perform iterative minimisation that considers all views simultaneously but rely on *mutual* correspondences; matches are defined between points that are *mutually nearest neighbour* and appropriate view transforms are found by employing GPA to find solutions that minimise the distance between mutual neighbours. The work of [262] shares the opinion of this thesis that considering all views simultaneously benefits overall result quality; however we propose a novel strategy on how point correspondences should be considered and handled when large numbers of view points are utilised (see Chapter 3 for further detail).

The work of Toldo et al. is theoretically sound, based on the well-known GPA theory and gives an efficient and elegant method to automatically align views in an ICP framework. The authors show experiments demonstrating their method's effectiveness for multi-view problem instances. A variant of the method, where point correspondences are non-uniformly weighted (using curvature similarity) is also presented. The approach can be applied in any case where the alignment of multiple views is required to be automatically refined and the algorithm is able to reach a global minimum even when scan pre-alignment is only roughly defined. Furthermore, the approach exhibits superior accuracy in every experiment (conducted in [262]) compared to the baseline technique of employing a classical ICP to multi-view datasets sequentially. The recent approach presented in [262] can be considered among the state-of-the-art methods for true *simultaneous* multi-view registration, and we thus consider the work of Toldo et al. to be a suitable candidate to compare the novel registration work introduced in this thesis with (see Chapter 3 for further detail).

A summary of multi-view registration algorithms is provided in Table 1. A broad array of alignment constraints, point correspondence rules and spatial transform finding methods have been proposed. We find that only a subset of the surveyed literature possess the desired properties and attempt the specific problem instances that we aim to explore further in this thesis (*e.g.* [210, 263, 260, 27, 88]). In the following chapters we predominantly compare the registration work introduced in this thesis (quantitatively and qualitatively) with: a baseline sequential ICP approach, the work of Pulli [210] and

the work of Toldo et al [262]. The work of [210], although now dated, meets all of the criteria we aim to investigate and the algorithm still proves a popular choice for current practitioners (*e.g.* the popular *Scanalyze* implementation [227]) due to the well understood methodology and favourable registration results for large view sets. Commonly, these attributes enable additional works (that also meet our problem instance criteria [263, 88, 26]; *c.f.* Table 1) to compare registration results directly with the methodology of [210].

As noted, the work of Torsello et al. [263] is similar in spirit to that of Pulli [210] and is shown to exhibit similar registration results in noise level experiments for all explored error metrics except *translation error* ΔT (see [263] for detail). Additionally the technique refines motion using pairwise ICP registration by alternating between 10 steps of ICP and their novel diffusion process until convergence. In [263] it is conceded that this pairwise registration alternation strategy clearly helps to avoid local minima in their experiments (but does not incur a noticeable penalty in running times due to their frugal diffusion process). For this alternation strategy to be utilised, unlike [262] and the work introduced in this thesis, pairwise view knowledge is required, potentially making multi-view registration not achievable if none is available. It should additionally be noted that Torsello et al. [263] and Bonarrigo et al. [26] do not optimise correspondences, which are considered fixed or allowed to vary in alternation with the optimisation of the rigid transformations (but the convergence of such procedures is not discussed). Due to the noted registration result similarity, and [263, 26] comparing directly with the work of [210], in this thesis we perform direct experimental comparison with [210] thus providing a common experimental test-bed yet also facilitating qualitative proxy-comparison with more recent methods. Future work may utilise additional implementations or datasets to facilitate further direct methodology comparison (*e.g.* with [263, 26]).

By extending the pairwise LM-ICP registration framework of Fitzgibbon [94] to multi-view, the work of [88] introduces an additional true *simultaneous* view optimisation strategy independent of pairwise registration information and utilises sculpture and statue themed datasets; *Capital* (100 views), *Madonna* (170 views) and *Gargoyle* (27 views). In practice quantitative registration results using an *average registration error* (mm) are only reported for *Bunny* (10 views) and *Gargoyle* (27 views) datasets due to their reference baseline implementation (the Pulli Scanalyze [210, 227] system) crashing for any datasets larger than 30 views (see [88] for detail). Since results that would allow

quantitative comparison of large view-set datasets are not available in the original work of Fantoni et al. [88], in this thesis we decide not to evaluate direct comparison with the methodology of [88], however their registration error (mm) results on small view-set data are observed to produce a *c.* 2% performance improvement in comparison to the implemented *Scanalyze* system [227, 210] (see [88] for detail). By performing our own quantitative comparison with the *Scanalyze* system [227] we again allow for comparison-by-proxy to [88], for small view-set cases.

By evaluating our work directly with the methodology of [262] we provide comparison to additional work that does not require pairwise view-set knowledge (*c.f.* [210, 263]). Such methodology is applicable to situations where no pairwise information is available. We note that the recent method of Toldo et al. [262], although not applied to large view sets in the original work, provides state-of-the-art multi-view registration performance in instances where no pairwise information is required and the method also proves amenable to large view set experimentation in practice. Furthermore, an additional example of a true *simultaneous* multi-view registration approach provides a logical and challenging comparator to the work introduced in this thesis.

As will be alluded to in section 2.4, it can be difficult to quantitatively compare registration methods in terms of registration quality due to factors such as: diversity of metrics used, experimental conditions, differences in hardware and software and the large heterogeneity existing in the data sets considered (*e.g.* creation/acquisition tools and equipment, synthetic/real (noisy) data, point/vertex density, mesh/point cloud format, number of views, etc). In [27] it is noted that optimisation convergence trend studies are valuable but should be carefully verified on real-world data. As discussed in their work, it is not guaranteed that a better *rate of convergence* is also always towards a *better minimum*, in the case of (for example) multi-view registration problem instances. A somewhat sparse and disparate coverage in the literature leaves multi-view registration techniques suffering from a lack of robust and fair methodology for performance assessment and comparison. It can be difficult in real scenarios to evaluate and quantify the results of a global alignment and determine the best solution without resorting to a thorough and time-consuming qualitative analysis of registered views. In the work presented in this thesis we provide additional methodology towards one route to tackle a lack of ground truth availability in real world scenarios (see Chapter 5, section 5.4.1.2).

Table 1: Characteristics of a selection of prominent multi-view registration algorithms

Method	Accom. *large* view sets	Accom. generic topology	Global optim-isation	Alignment constraints	Transformation computation
Robust GA (*Silva et al.* [238])	–	✓	–	interpenetration measure	genetic algorithm
Signed distance field matching (*Masuda* [172])	–	✓	–	point-point	distance minimisation
Geometric model generation (*Masuda* [171])	–	✓	–	point-point	distance minimisation
ICP with rand. sampling (*Masuda and Yokoya* [173])	–	✓	–	chained pairwise	distance minimisation
Point-to-plane reg. (*Chen and Medioni* [54])	–	✓	–	chained pairwise	distance minimisation
Simulated Reannealing reg. (*Blais and Levine* [25])	–	✓	–	control point sampling	stochastic optimisation
View network reg. (*Bergevin et al.* [19])	–	✓	✓	view-pairs network	linear least-squares
Large view set reg. (*Pulli* [210])	✓	✓	✓	pairwise constraints	pairwise constrained optim.
Auto. model building (*Gagnon et al.* [103])	–	✓	✓	point-plane	distance minimisation
Quadratic cost function reg. (*Williams and Bennamoun* [279])	–	✓	✓	generalised ICP	distance minimisation
Unit quaternion reg. (*Benjemaa and Schmitt* [15])	–	✓	✓	quaternion decoupling	distance minimisation
Global reg. with multi-z buffer (*Benjemaa and Schmitt* [16])	–	✓	✓	multi-z buffer	distance minimisation
Mean rigid shapes (*Pennec* [201])	–	✓	–	mean rigid shape	distance minimisation
Physically inspired reg. (*Stoddart and Hilton* [249])	–	✓	✓	phys. inspired energy	Euler method solving dyn. sys.
Simulated springs reg. (*Eggert et al.* [84])	–	✓	✓	force-based energy	iter. motion computation
Globally consistent sub-graphs (*Huber and Hebert* [131])	–	✓	✓	point-plane	global optim. graph search
Least-squares distance reg. (*Neugebauer* [189])	–	✓	–	point-plane	iter. least-squares dist metric
Large planar surface reg. (*Pathak et al.* [198])	✓	–	✓	plane-based	pose-graph relaxation
Reg. using planar features (*Previtali et al.* [208])	✓	–	✓	planar features	linear least-squares
Coordinate frames (*Sharp et al.* [235])	–	✓	–	optim. inter-frame graph cycles	optim. over neighbouring view graph
Frame space reg. (*Sharp et al.* [234])	–	✓	–	inter-frame graph cycles	optim. over graph cycles
Dynamic Geometry reg. (*Mitra et al.* [184])	✓	✓	–	space-time surfaces	linear system
Gen. procrustes analysis (*Toldo et al.* [262])	–	✓	✓	procrustes analysis	generalized proc. analysis
Graph diffusion (*Torsello et al.* [263])	✓	✓	✓	dual quaternion	view-graph diffusion
Rank min. reg. (*Thomas and Matsushita* [260])	✓	✓	✓	rank minimisation	convex optimisation via Lagrange Mult.
Manifold optim. (*Krishnan et al.* [153, 154])	–	✓	✓	unconstrained optim.	optim. on a manifold
Improved manifold optim. (*Bonarrigo and Signoroni* [27])	✓	✓	✓	unconstrained optim.	optim. on a manifold
Kinematical analysis reg. (*Pottmann et al.* [206])	–	✓	✓	first ord. kinematics	iter. compute instantaneous motion
Geometric optimisation reg (*Pottmann et al.* [207])	–	✓	✓	geometric optimisation	Gauss-Newton
Multi. LM-ICP (*Fantoni and Castellani* [88])	✓	✓	✓	generalised LM-ICP	Levenberg Marquardt

A summary of characteristics for a number of prominent multi-view registration techniques. We define *large* view sets as documented experimentation with ≥ 100 views. Additional techniques considered in this chapter do not provide an end-to-end *multi-view registration* solution for point sets (*e.g.* they focus instead on multi-view point set *integration*). Since the focus of this thesis is predominantly multi-view registration, such techniques have not been included in this table.

In this thesis we propose a novel simultaneously multi-view registration technique where view registrations are found through the optimisation of an energy measure defined over the points of the input scans. With every point of the input data we associate a local measure capturing the likelihood that the point is located on the underlying sampled surface. Using kernel density estimation, a fundamental data smoothing technique, we make inferences about where surfaces exist based on the data samples available and use these inferences to align scan views by gradient ascent optimisation over the pose space parameters. Following pose space optimisation we then refine our kernel density model estimate iteratively in a similar global strategy to many of the techniques highlighted in this chapter. The rationale is that since only a limited number of points are sampled from the true surface, the position of every surface point is partly uncertain. By capturing this fact in our density estimation approach we are able to exhibit robustness in the presence of sensor noise and scan misalignment. By optimising parameters in the transform space with respect to our energy function we reduce the amount of registration work required since no pairwise point correspondences, view pair scan alignment or view order information is required, as is commonly the case in the surveyed previous work. By requiring neither prior knowledge of view order nor of individual point correspondences the proposed framework is simple, automated and theoretically sound. In the following section we consider computational issues when working with large view set point cloud data.

2.3 LARGE VIEW SET CONSIDERATIONS

2.3.1 *Global optimisation for large view set registration*

As surveyed in section 2.2, several heuristic methods have been proposed to handle the global multi-view registration problem. One emerging alternative strategy to global multi-view registration involves numerical optimisation (see *e.g.* [150, 157, 151, 152]). Global registration has been tackled by several techniques that include the formalisation and solving of non-convex minimisation problems with constraints related to the rigid transformations of point sets belonging to the views. The registration work proposed in this thesis follows such a strategy (see chapter 3 for further detail). Kolev et al. introduced a global optimisation method utilising a continuous convex relaxation scheme.

Specifically, the authors propose to cast the problem of 3D shape registration as one of minimising a spatially continuous convex functional. This style of approach often provides several benefits such as not requiring (the typically expensive) direct point-pair correspondence search during iterative registration; however, common weaknesses of optimisation based approaches must be carefully addressed. Large optimisation problem instances involving non-linear objective functions, expensive function evaluations, large numbers of optimisation variables (or combinations thereof) can often generate computational issues. The global nature of large view-set, dense point cloud registration problem instances provide one such area where optimisation methods need to be well designed and conditioned in order to reduce the risk of being stuck in local minima or to otherwise behave inappropriately.

2.3.1.1 *Point cloud sub-sampling*

Updating point correspondences iteratively as spatial transforms are improved typically brings an increase in cost proportional to the point correspondence count n (and view-set size) considered. When considering large view-sets, this cost is generally not appealing for practical applications if algorithm space and time requirements result in impractical runtime. A common solution to aid iterative updates for large n is found by down-sampling point sets (see section 5.4.4.1 for detail of sub-sampling utilised in this work. Methods to reduce the size of point set samples include feature point extraction and re-sampling [57, 58, 72], uniform point sub-sampling, feature point extraction and fusion [59] and image point decimation [110]. These techniques are often used to reduce the number of points in free-form shapes for feasible registration of large point sets and other tasks. In practice uniform sub-sampling may be sufficient in many cases yet if user-designated feature-sensitive sub- and re-sampling is required then more advanced methods are likely to be required (see [199] for a comprehensive review of simplification methods).

2.3.2 *Point correspondences for large view sets*

Rather than exploiting the same point correspondence set throughout the optimisation process, a common alternative involves updating correspondences while a registration strategy iteratively refines viewpoint alignment. By choosing to update correspondences

iteratively, registration accuracy may be improved, potentially at some computational cost when a typical measure is used (*e.g.* closest point distances) to find correspondences. Correspondence updating (and related registration accuracy improvement) tends to come at the price of heavier computational cost. Increasing view-set count naturally also increases point correspondence counts. Methods that strike a balance between accuracy and speed are highly sought-after for practical applications. Updating point correspondences at each iteration can be considered [153] unappealing from a cost perspective as this is often a prime contributing factor to method computational expense [21]. Approaches that update point correspondences iteratively are however advantageous as (providing that a reasonable initial alignment is offered) each iteration will bring the views closer to an acceptable solution. This in turn improves the correctness of the next correspondence set until convergence is reached. The registration framework proposed in this thesis attempts to harness the noted advantages of continuous iterative assessment and evaluation of individual point positions without requiring direct point pair correspondence updating (see chapter 3 for further detail).

2.3.2.1 *Soft point correspondences*

Related to correspondence *updating* choices are the *type* of correspondences made use of. Pairwise correspondence work [106, 110, 159] has progressed the state-of-the-art of two-view registration by replacing *hard* point pair correspondences with *soft* correspondences. This typically involves each point in one point set corresponding somehow to every point in the other set by some weight similar to the probabilistic techniques outlined in section 2.1. Robustness to a wide basis of initial coarse misalignment is a desirable property, when it is recalled that the condition of a reasonable initial coarse alignment is required for the strategy of iteratively updating correspondences in an attempt to converge to an optimal registration. The associated computational cost of these *soft* correspondence methods has however prevented their practical usefulness, even for moderately large point sets in the two-view case.

Related work facilitates registration of large sets of unstructured point clouds by opting not to use point correspondences at all (*e.g.* [184] use kinematic properties of space-time surfaces to solve for alignment motion). The work of [184] was also shown to provide better results than traditional ICP based approaches in terms of handling large ranges of initial coarse alignment. In [184] objects are scanned at high frame rates and

large viewpoint sets are handled by avoiding the explicit computation of point correspondences between successive frames altogether. By noting that inter-frame motions are generally small the underlying temporal data coherence is exploited and several frames can be integrated at once to directly compute object motion from scan data. In a further effort to make computation more frugal, [184] avoid performing any form of global relaxation, noting that this becomes expensive for large view sets. By computing the smoothness of the underlying space-time surface they avoid computing corresponding points between successive frames however, by neglecting any form of global error distribution, the method may still be susceptible to alignment error accumulation.

Additional recent extensions attempt to combine the noted robustness and accuracy benefits that *soft* correspondences afford with fast execution time in an effort to improve practical usefulness. By attempting to combine *soft* correspondences with the efficiency of traditional ICP style approaches [158] or a General-Purpose-GPU (GPGPU) based implementation [253], *soft* correspondence based registration is becoming more feasible. Additionally, recent alternative methods, used to reduce computational burden, have been found by employing a more sophisticated correspondence update procedure such as [27] where sub-sampling of correspondences is used to avoid expensive rototranslation and matching of entire point sets.

2.4 REGISTRATION AND RECONSTRUCTION QUALITY EVALUATION

Evaluating multi-view scan registration quality (and following pipeline component quality *e.g.* surface reconstruction) is an active area of investigation where performing rigorous and illuminating evaluation can be considered a challenging task in it's own right. When attempting to evaluate competing method output, it is often not sufficient to perform only visual comparison among strategies. Results of differing methodology can look visually reasonable while containing varying sets of potentially subtle or difficult to perceive registration or reconstruction flaws. One of the issues that make assessment challenging is that it becomes difficult to perform quantitative evaluation without a known object ground truth model or surface. Since such models are not always available, evaluation techniques can be categorised into two groups; according to whether or not known ground truth is a requirement.

In cases where ground truth is not available evaluation tends to concentrate on consistency between method results and partial scan input data. Typical evaluation metrics for the (post-registration) model reconstruction task have included *reconstruction errors* [8]; the average Euclidean distance from input points to a reconstructed surface and, in a similar vein, *integration errors* [287, 288]; calculating the mean Euclidean distances between points in a final reconstructed surface and their closest corresponding points in the input data viewpoints. Reconstruction accuracies have also been quantified by measuring a mean per-point distance of registered range data to reconstructed surfaces [283]. A recent method that takes into account global reconstruction and local registration detail is the method of 3D Gini Coefficients (3DGiC) introduced in [245]. The 3DGiC is a metric that depends on both global consistency and local accuracy of registration in order to deliver an evaluation based on cumulative distributions of local surface descriptors.

When a complete ground truth model is available, it can easily be used as part of the direct evaluation of registration and surface reconstruction results. A range of methods have been implemented for this purpose including employing mean square errors [270] of reconstructions against a known ground truth for comparison and measuring reconstruction standard deviations to a ground truth under varying levels of noise [129]. A metric named the *shape error* [138, 205] can be calculated using the ratio between the volume of the symmetric difference between an estimated surface and the ground truth and the volume of the ground truth. To measure the *accuracy* of a reconstruction the authors of [231] begin by calculating the signed distances between the points in the reconstructed model and the closest corresponding points of the ground truth model. This technique outputs a single distance value such that 90% (author suggested level) of the reconstruction is within the distance threshold of the ground truth model. In the work of [18] it is proposed that a tri-step method is used to evaluate reconstruction error where the ground truth models were produced via a commercial optical laser scanner.

In conclusion evaluation methods for multi-view registration and reconstruction can often be thought of as measures of registration *tightness* and shape *dissimilarity*. A good measure can be thought to satisfy some general requirements [194] desirable for dissimilarity metrics and some specific properties that can be considered useful for the particular applications considered in this thesis. In [245] examples of good evaluation metric properties are discussed and include *invariance* to rigid transform, *robustness* to

small perturbation, *generality* such that a variety of input data can be accommodated and *applicability* providing a measure with means to be utilised in multiple scenarios (this typically becomes possible when a ground truth model is not a requirement). As discussed here, there is a body of work detailing surface reconstruction evaluation however fewer multi-view registration specific metrics are found. In the following chapter (see 3.5) we introduce novel registration specific metrics by taking into account the desirable properties considered in this section that allow us to compare multi-view registration results both quantitatively and qualitatively.

2.5 DISTRIBUTED COMPUTATION

Distributed computation is explored in this thesis to improve the runtime of computationally demanding registration. Distributed compute clusters allow the computing power of heterogeneous (and homogeneous) resources to be utilised to solve large-scale science and engineering problems. One class of problem that has attractive scalability properties, and is therefore often implemented using compute clusters, is task farming (or parameter sweep) applications [239]. A typical characteristic of such applications is that no communication is needed between distributed subtasks during the overall computation. However interesting problem instances have also been formulated under large-scale task farming such that global communication between subtask sets take place [282]. This allows the formulation of problems that contain subtasks possessing both independent and synchronised elements. This thesis explores these problem formulation strategies applied to the problem of global multi-view point cloud registration.

Employing multicore processors to parallelise the task of 3D point cloud registration in particular has been recently investigated [170] by extending a previously introduced coarse binary cubes registration approach. In contrast to the work presented in this thesis, Martinez et al. perform parallel evaluation of prospective transform solutions in a globalised Nelder-Mead search whereas this thesis explores multicore parallelism at a granularity that distributes the registration of an *entire point cloud* (viewpoint) per core. In chapter 4 we propose a framework called *semi-synchronised task farming* in order to address the global simultaneous view registration problem feasibly for very large point cloud view-sets in problem instances with time constraints. We propose to handle global communication between task sets with a post task set completion synchronisation

step, following a round of concurrent computation. Our framework is inspired by the influential Bulk Synchronous Parallel (BSP) model [267]. The BSP model of parallel computation is originally defined as the combination of three attributes; (1) A number of components, each performing processing and/or memory functions; (2) A router that delivers messages point to point between pairs of components; and (3) Facilities for synchronising all or a subset of the components at regular barrier intervals (see Figure 7).

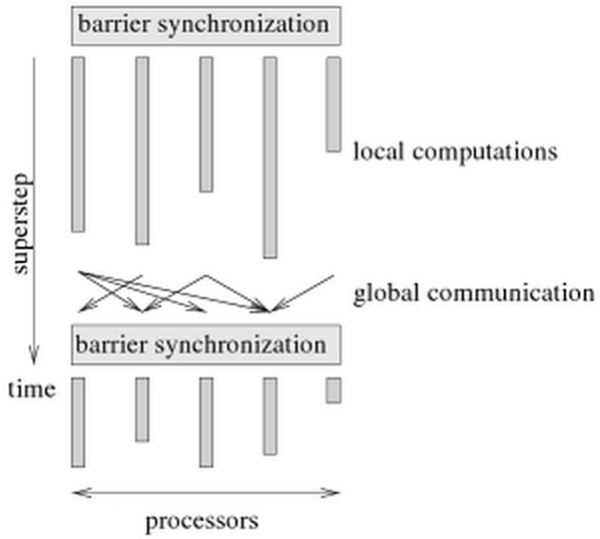


Figure 7: In the BSP model, computations are performed in supersteps where each superstep consists of three phrases (1) simultaneous local computations of each processor, (2) communication operations for data exchange between processors, and (3) a barrier synchronisation to terminate the communication operations and to make the data sent visible to the receiving processors. Figure adapted from [216].

There have been a number of previous general BSP library implementations, for example the Oxford BSP Library [125], Green BSP library [108], BSPlib [126] and Paderborn University BSP library [28]. They vary in the set of communication primitives provided, and in how they deal with distribution issues such as reliability (machine failure), load balancing, and synchronisation. The scalability (and fault-tolerance) of such BSP implementations has not however been evaluated beyond several dozen machines.

In this thesis we apply a task-farming framework, inspired by the BSP model, to the multi-view registration problem thus providing a novel parallelisation strategy for point cloud registration. By predicting time savings that our framework provides in simulation, and validating these predictions on our chosen application in practice, we

are able to reliably estimate performance gains obtained when using a BSP framework to tackle resource intensive registration tasks distributed to many cores. The usefulness of these runtime performance predictions when utilising hundreds of processing cores and the generalisability of the introduced BSP framework to a range of additional complex problems, drawn from real-world computer vision tasks, was recently explored in [178].

2.5.1 *Task farming*

The task farming model of high-level parallelism has been the basis for much HPC cluster based work with recent examples utilising HT Condor [257], Google's MapReduce [68] and Microsoft's Dryad [134]. The HT Condor framework is able to harnesses idle cycles from both a network of non-dedicated desktop workstation nodes (cycle scavenging) and dedicated rack-mounted clusters. The framework then employs these cycles to run coarse-grained distributed parallelisation of computationally intensive tasks. Task farming is also common in data centres, for example MapReduce and Dryad both make use of task farming to schedule parallel processing on large terabyte scale datasets. In systems such as these a master process manages the queue of tasks and distributes these tasks amongst the collection of available worker processors. The master process is typically also responsible for handling load balancing and worker node failure. In the current work, master and worker node interaction is handled by Sun Grid Engine (SGE) [105] using a batch queue system similar to the Condor framework. This queueing system is responsible for accepting, scheduling and managing the distributed execution of our parallel tasks. This approach allows the distribution of arbitrary tasks as there is no requirement for a specialised API.

Dedicated parallel computer architecture has also been employed to develop computer vision systems. In [217] a Beowulf architecture dedicated to real-time processing of video streams for embedded vision systems is proposed and evaluated. The parallel programming model made use of is based on algorithmic skeletons [61]. Skeletons are higher-order program constructs that encapsulate common and recurring forms of parallelism to make them available to application developers. Skeleton-based parallel programming methodology offers a partially automated procedure for designing and implementing parallel applications for a specific domain such as image processing. An

application developer provides a skeletal parallel program description, such as a task farm, and a set of application specific sequential functions to instantiate the skeleton. The system then makes use of a suite of tools that turn these descriptions into executable parallel code. The system in [217] was tested by implementing simple image processing algorithms such as a convolution mask and Sobel filter.

In comparison to classical HPC applications, embedded computer vision on dedicated parallel machines will often be able to offer advantages such as mobile, real-time performance yet places demands on programmers if no high-level parallel programming models or environments are available such as skeletons or the SGE that we make use of in this work (see chapter 4 for detail). If these tools are not available then programmers must explicitly take into account all low-level aspects of parallelism such as task partitioning, data distribution, inter-node communication and load balancing. If developer expertise lies in (for example) image processing, rather than parallel programming, then accounting for these low-level considerations likely results in long and error-prone development cycles.

In contrast to [217] in this thesis we perform task farming as opposed to low-level data parallelism involving geometric partitioning of images for image processing tasks. This results in a coarser level of abstraction that we apply to high level computer vision problems involving large data sets such as the discussed large-scale point cloud registration problems. It is for this reason that we consider the BSP model a good basis for our framework. The original BSP model considers computation and communication at the level of the entire program. The BSP model is able to achieve this abstraction by "*renouncing locality as a performance optimisation*" [242]. This in turn simplifies many aspects of algorithm design and implementation and does not adversely affect performance for most application domains. Low-level image processing however is an example domain for which locality might be critical so a BSP based framework is likely not the best choice there.

Parallel and distributed computing systems are designed with performance in mind and significant previous work has been carried out developing approaches for performance modelling and prediction of applications running on HPC systems. In addition to the BSP inspired framework in this thesis we formulate a performance model allowing the prediction of run time performance of the parallel algorithms implemented within the framework. Application performance modelling involves assessing application

performance through system modelling and is also an established field [116]. Several examples of where this approach has proven advantageous include: input and code optimisation [187], efficient scheduling [246] and post-installation performance verification [146]. The process of computational modelling itself can be generalised to three basic approaches; modelling based on analytic (mathematical) methods, (*e.g.* LoPC [100]), modelling based on tool support and simulation (e.g. DIMEMAS [155], PACE [191]), and a hybrid approach which uses elements of both (*e.g.* POEMS [4], Performance Prophet [202]). In this thesis we choose a hybrid approach and combine basic analytical modelling inherited from the BSP model with traditional code profiling. Details of our performance modelling approach are provided in chapter 4, section 4.3.3 and the approach is then applied to point cloud registration problems, with a detailed performance analysis to demonstrate framework scalability in section 4.4. We apply our BSP inspired framework to large-scale depth image and point cloud registration in chapter 5 and explore the computational benefits this strategy is able to afford using both synthetic and real large view-set registration problem instances.

2.6 SUMMARY

In this chapter we have highlighted the established history of image registration and surveyed a number of existing approaches for pairwise and multi-view point cloud registration in the literature. Most of the techniques incorporate distance minimisation in one form or another - typically with the use of iterative point pair correspondence association. Such models iteratively associate point pairs over the set of viewpoints to be aligned allowing registration algorithms to find sets of suitable spatial transforms facilitating convergence to good view alignment. Many ICP [21] variants exist and typically provide modifications to the point matching strategy. However, such low-level pairwise statistics are often not enough to provide good results in the multi-view case and therefore various ad-hoc multi-view strategies have been proposed. State-of-the-art techniques additionally incorporate *soft* point pair correspondence [106, 253] evaluation to provide improved tolerance to noisy data and bad initial coarse view alignment at the cost of additional computational expense. In the case of multi-view registration, the majority of state-of-the-art registration techniques select one of the point sets as the "model" and perform pairwise alignments between the other sets and this set. A

drawback of this mode of operation is that there is no guarantee that the model-set is free of noise and outliers, which contaminates the estimation of the registration parameters. Unlike previous work, the proposed method treats all point sets on an equal footing: they contribute to a kernel density estimation and the task of finding optimal alignments is cast as an optimisation problem.

Computational cost becomes an important factor when considering the large view-set data offered by contemporary high frame rate depth sensors and various routes exploiting parallelism at assorted granularities are emerging to address this. As discussed, a registration technique's efficacy is determined by its accuracy and the computational complexity of its component algorithms with a suitable balance between accuracy and speed being a generally sought-after trait for modern practical applications making use of 3D point cloud data.

Part III

MULTI-VIEW REGISTRATION USING DENSITY ESTIMATION

3

MULTI-VIEW REGISTRATION USING DENSITY ESTIMATION

3.1 INTRODUCTION

In this chapter we present our point set registration and implicit surface approximation framework based on density estimation. We show how this framework can be used to solve the *multi-view* registration problem enabling the robust registration of multiple sets of points representing depth measurements sampled from varying viewpoints of object surfaces. Our approach to density estimation is non-parametric, and provides an implicit surface estimate using the available point data as evidence. By iterating between updating this estimated surface shape and improving the alignment of individual viewpoints in relation to our surface estimate, we bring all partial views into globally consistent alignment. We apply this approach to 3D surface data, represented by multiple dense point clouds, where we assume that point correspondences between scans and view order are initially unknown. Given many partial 3D data sets, typically captured by active or passive depth sensors, from differing viewpoints, our density estimate approximates the underlying sampled surface and using this surface estimate we define an energy function that implicitly considers the spatial position of all partial viewpoints simultaneously. We use this density estimate to guide an energy minimisation in the transform space, aligning all partial views robustly.

Given many partial object views, we estimate a density function of the point data to determine an approximation of the sampled surface. With every point of the input data we associate a local kernel capturing the likelihood that the 3D point is located on the sampled surface using neighbouring points as evidence. This measure takes into account the normal directions estimated at the scattered points. Using this density function we employ an energy minimisation strategy that implicitly considers all view-

points simultaneously. Using the density estimate we guide an energy minimisation in the transform space, aligning all partial views robustly. We evaluate this strategy quantitatively on synthetic and range sensor data where we show improved robustness and registration accuracy through comprehensive experiments that compare our approach with a selection of the main competing frameworks for this task.

Our experiments on a variety of data sets demonstrate the advantages of our kernel density registration approach: First, we show that our density estimate is capable of accurately representing object surfaces robustly, and demonstrate that these surface estimates are accurate enough to be used for the task of point set registration. Second, we apply our kernel density registration approach to the *multi-view* registration problem, and show that performance is better than the state-of-the-art on a number of benchmark data sets.

In this chapter we introduce our non-parametric kernel density estimation technique to address the point cloud registration problem. After briefly introducing the elementary components of the canonical kernel density estimation technique, we provide detail on the importance of bandwidth parameters, how these may be selected and the role they play in the problem instances addressed in this work. We then go on to describe our framework for addressing point cloud registration problems utilising kernel density estimation. We document experiments that provide evidence that this approach gives improved performance when solving point cloud registration problems. We conclude the chapter with some discussion and suggestions for future work.

3.2 DENSITY ESTIMATION

Density estimation techniques provide a set of tools for constructing an estimate of an unknown density function, based on observed data. The unknown density function represents an underlying density according to which a large population is distributed. The observed data points are usually thought of as a random sample from that population. Density estimation techniques can be split into those that make assumptions about the form of the density in question, parametrised in some way (parametric estimation), and, alternatively, those that avoid making such assumptions about the form of the underlying distribution (non-parametric statistics). The approaches described in this chapter make use of various tools and concepts from the large body of work that con-

cerns non-parametric density estimation. In the following sections related foundational concepts are briefly outlined.

3.2.1 *Non-parametric density estimation*

When high accuracy or assumption-free density estimation is a requirement, non-parametric methods are often an appropriate choice. The general formulation of non-parametric density estimation is an approach to approximate a density function $f(x)$ at any point x without making assumptions about the form of f. The key idea involves independently looking at each point x at which we want to approximate the density and then deciding which of the available data observations $p_1, \ldots, p_N$ should be used to estimate $f(x)$ at x. This is typically done by only taking data observations in a *neighbourhood* around x into consideration. In such cases only the observations contained in a specified interval are used to approximate the density $f(x)$ at x.

A simple example in the one dimensional case involves considering data observations contained in an interval of length 1 centred at x. In this case all observations p_i where the absolute difference from x is greater than 0.5 are ignored in the estimation.

In general if x is a point at which the density function is to be estimated, then A is a *neighbourhood* such that a data observation p_i contributes to estimate the density $f(x)$ at x if and only if p_i is contained in A. For N observations $p_1, \ldots, p_N$ the density function at x can then be approximated as:

$$\hat{f}(x) \cong \frac{k}{NV}$$

where V is the volume of the *neighbourhood* A considered and k denotes the number of data observations contained in A. The general formulation for non-parametric density estimation can then be found as follows: consider the probability that a point x, that has been drawn from the distribution $f(x)$, falls into a region A of the sample space. Call this probability:

$$P = \int_A P(t)\,\mathrm{d}t$$

Then for a set of N data points, the probability that exactly k of these fall into the region A can be modelled using a binomial distribution:

$$P(k) = \binom{N}{k} P^k (1-P)^{N-k}$$

From the properties of the binomial probability mass function we can estimate the mean and variance of $\frac{k}{N}$ as:

$$E\left[\frac{k}{N}\right] = \frac{1}{N} E\left[k\right] = P \qquad Var\left[\frac{k}{N}\right] = \frac{1}{N^2} Var\left[k\right] = \frac{P(1-P)}{N}$$

As N increases the variance will decrease so in the limit, as N tends to infinity, a good estimate of P is:

$$P \cong \frac{k}{N}$$

However, if it is also assumed that region A is so small that the density function, $f\left(x\right)$, does not vary within A, then the probability that x is in A is given by:

$$P = \int_A P(t)\,\mathrm{d}t = P(x)V$$

where V is the volume enclosed by A. By combining these two results for P, the general form of non parametric density estimation is reached:

$$P(x) \cong \frac{k}{NV}$$

To improve how accurate this estimate of $P\left(x\right)$ is, the size of V should approach zero. However if the size of V gets too small the volume will not enclose *any* of the observed data points. Therefore, a good compromise must be found, such that V is large enough to include data points but small enough to give a good estimate of the probability $P\left(x\right)$.

3.2.2 *Generalisation*

One disadvantage of simply using intervals to define the data point contributions to the density estimate is that if an observation is in the interval, the distance between x

and the observation is not taken into account. This may lead to a poor approximation of the density function at x. For example, when estimating the density at x, if there are many observations at the edge of the interval but only a few very close to x, this would not be taken into consideration when estimating the density. It may therefore be reasonable to give a stronger weighting to observations *closer* to x than those that are farther away.

To determine the significance of an observation p_i when estimating the density function at x, a non-negative function $G(x, p_i)$ is defined. In the previous formulation above, $G(x, p_i)$ would equal 1 if observation p_i was in the unit interval and otherwise $G(x, p_i)$ would be equal to 0, corresponding to the observation being ignored in the estimation of x.

In the general case, for point x, the function $G(x, p_i)$ for $i = 1, \ldots, N$ determines the significance of the observation p_i in the estimation of the density at x. The density function at x is then approximated as:

$$f(x) \approx \frac{k(x)}{N \cdot V(x)} \tag{2}$$

where

$$k(x) = \sum_{i=1}^{N} G(x, p_i) \qquad V(x) = \int_{\mathbb{R}} G(x, y)\, \mathrm{d}y$$

3.2.3 *Kernel density estimation*

Kernel density estimation constitutes a set of classical non-parametric density estimation techniques that date back to [218] and [197] and reside in the framework outlined above. Estimating density functions with similar non-parametric techniques (*e.g.* histograms), can result in density estimates that are not smooth and estimation accuracy performance can be influenced by *e.g.* the choice of histogram bin start and end positions. Using statistical kernels for density estimation attempts to solve these shortcomings. For example, through the choice of a suitable kernel, the estimation provided can be endowed with properties such as smoothness and continuity.

The kernel density estimation approach involves taking the general density estimation formulation outlined previously and firstly fixing the volume V to be 1 for all x

and secondly defining the general formulation function $G(x, p_i)$ as a one-dimensional function $K(x - p_i)$ that only depends on the absolute difference between x and p_i and not on the actual data point values. This implies $G(x, p_i) = G(p_i, x)$.

Let u denote the difference $x - p_i$. Then the function $K(u)$ is symmetric, non-negative and has integral 1. The function is symmetric since $K(x - p_i) = K(p_i - x)$ and non-negative since the general formulation function $G(x, p_i)$ is non-negative. A statistical kernel is a non-negative, symmetric function centred at zero with integral 1. Therefore we can view $K(u)$ as a kernel. By reconsidering the general formulation of a non-parametric density estimate (Equation (2)), fixing the volume $V(x)$ to 1 and defining $k(x)$ in terms of this statistical kernel:

$$k(x) = \sum_{i=1}^{N} G(x, p_i) = \sum_{i=1}^{N} K(x - p_i)$$

we can approximate the density function at x using the kernel K as:

$$f(x) \approx \hat{f}(x) = \frac{1}{N} \sum_{i=1}^{N} K(x - p_i) \tag{3}$$

3.2.4 *Popular kernel choices*

The choice of kernel function influences the effect that observed data points have on the estimation at point x. Popular choices of kernel functions include: uniform, Epanechikov and Gaussian kernels with further options including triangular, biweight, triweight, tricube and cosine kernels. Figure 8 gives an illustration of the kernel density estimation technique for scattered data samples ('+' crosses) in one dimension. Local maxima of the density estimation $\hat{f}(\cdot)$ (black line) naturally define clusters in the scattered point data. The kernel choice in the example given is Gaussian (blue).

A uniform kernel takes value 1 if the absolute value of u is less than or equal to 0.5 and 0 otherwise. Note that the estimated density is not smooth since the kernel summation only takes integer values; the estimated density function will have discontinuities and takes a constant value on intervals of length 0.5. The density estimate is often prone to local noise in this case. Conversely, a Gaussian kernel never takes the value zero. Therefore a typical Gaussian kernel considers every observation when estimating the density function at point x, but observations close to x are weighted higher

than those further away. Density functions estimated using Gaussian (and other *e.g.* Epanechnikov) kernels are therefore smooth and typically produce smooth estimates as can been observed in Figure 8.

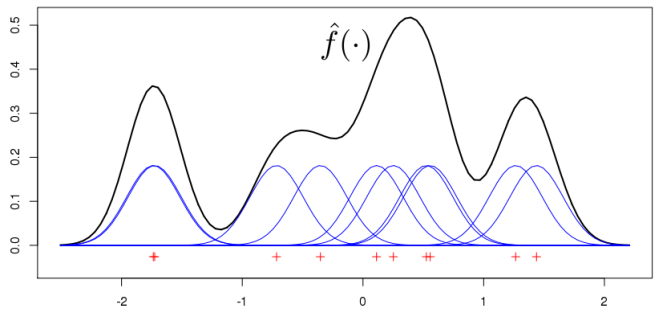


Figure 8: Example of the kernel density estimation technique for 1D data points. Data points are marked with the '+' symbol. Note that local maxima of the kernel estimation define clusters of the original data. The density approximation $\hat{f}(\cdot)$ (Equation (3)) at a given location $\mathbf{x}$ is the summation of values contributed by local Gaussian kernels K, located at each data point.

3.2.5 *Kernel bandwidth*

The discussed uniform kernel only considers observations with an absolute distance from x that is smaller than 0.5. In order to extend the *neighbourhood* of considered observations, one possibility is to transform the kernel. As an example, if we double the size of the *neighbourhood* considered, then the uniform kernel behaves as follows: if the absolute distance to x is smaller than 1, the uniform kernel takes the value 0.5; if the distance is greater than 1, it takes the value zero.

This change to the *neighbourhood* is typically captured by a *bandwidth* parameter $h > 0$. The significance of observation p_i can therefore be defined:

$$K\left(\frac{x - p_i}{h}\right)$$

Note that by including this bandwidth parameter h, both the summation of the kernel contributions $k(x)$, and the volume V, found in the general formulation (Equation (2)) are influenced. The volume V is determined by the fact that the integral of $K(u) = 1$. Therefore:

$$V = \int_{\mathbb{R}} K\left(\frac{u}{h}\right) \mathrm{d}u = h$$

Applying these considerations to the general formulation, we obtain the following approximation of $f(x)$ from N observations $p_1, \ldots, p_N$ as:

$$f(x) \approx \hat{f}(x) = \frac{k(x)}{N \cdot V(x)} = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{x - p_i}{h}\right) = \frac{1}{N} \sum_{i=1}^{N} K_h(x - p_i) \tag{4}$$

Equation (4) provides the standard kernel density estimate for univariate distributions where K_h indicates a kernel employing a scalar bandwidth h. The two parameters required by kernel density estimation have now been established: the kernel function and the bandwidth parameter. It is widely agreed that the selection of an appropriate bandwidth h is important. In practice, the choice of the kernel is not as important as the choice of the bandwidth. A theoretical background for this observation is provided by [169] who note that kernel functions can be rescaled such that the difference between two kernel density estimates using two different kernels is small. Implications of appropriate bandwidth choice are considered in section 3.2.7.

The following section briefly outlines extending kernel density estimation to the multivariate case and then the process of optimal bandwidth selection is discussed.

3.2.6 *Multidimensional kernel density estimation*

Multidimensional kernel density estimation provides a natural extension of these estimators to multivariate data. Due to work carried out during recent decades multivariate kernel density estimation has reached a level of maturity comparable to the univariate counterparts. In a similar fashion to the univariate case let $p_1, p_2, \ldots, p_N$ be a set of d-variate samples drawn from an unknown multivariate distribution with density function $f(x)$. The multidimensional kernel density estimate of $f(x)$ can then be defined as:

$$\hat{f}_{\mathbf{H}}(x) = \frac{1}{N} \sum_{i=1}^{N} K_{\mathbf{H}}(x - p_i) \tag{5}$$

where

$$K_{\mathbf{H}}(u) = |\mathbf{H}|^{-\frac{1}{2}} K\left(\mathbf{H}^{-\frac{1}{2}} u\right)$$

In the multidimensional case the bandwidth is now defined by a symmetric, positive definite $d \times d$ matrix $\mathbf{H}$. Similar to the 1D case this parameter set again dictates the *amount* of smoothing induced by the estimate but also now controls a smoothing *orientation* that was undefined in the case of univariate kernels.

Parametrising this bandwidth matrix typically follows one of three parametrisation classes. In increasing order of complexity these classes are:

- S the class of positive scalars times the identity matrix
- D diagonal matrices with positive entries on the main diagonal
- F symmetric positive definite matrices

The class of kernels defined by S bandwidth matrices have the same amount of smoothing applied in all coordinate directions, D matrices allow for varying amounts of smoothing in each dimension and kernels making use of F matrices allow for an arbitrary *amount* and *orientation* of smoothing in each dimension. Use of kernels employing S and D bandwidth matrices tend to be widespread due to computational reasons, but previous work has shown that gains in density estimation accuracy may be obtained when using kernels that utilise the more general F class of bandwidth matrix [77], affecting both the *size* and *shape* of the kernels used.

3.2.7 *Optimal bandwidth selection*

In both univariate and multivariate cases, selecting an appropriate kernel bandwidth is of great importance as this free parameter h (or matrix of parameters $\mathbf{H}$) typically exhibits a strong influence on the density estimate. The problem of selecting a scalar bandwidth in univariate cases can be considered well understood [78]. A selection of plausible optimisation methods exist that couple good theoretical properties with strong performance in practice (see [141] for a review). Many of these techniques can be extended to multivariate cases in a straightforward fashion if H is constrained to be a diagonal matrix (*c.f.* section 3.2.6 and [273], [224] for further detail). However, imposing these constraints on the bandwidth matrix may produce decidedly suboptimal density estimates, even in cases where data dimensions have been pre-scaled or pre-sphered ([272], [79], [78]).

Using a bandwidth that is too small will often result in an under-smoothed estimate that is likely to contain many spurious data artefacts. Conversely using too large a bandwidth can result in an over-smoothed estimate, leading to much of the underlying structure of the distribution being obscured. A bandwidth that is too small produces a large variance and a small bias whereas a bandwidth that is too large leads to a low variance and large bias. It is in this regard that bandwidth selection corresponds to balancing bias and variance. In some situations, it is sufficient to subjectively choose a smoothing parameter by looking at the density estimates produced by a range of bandwidths. However, as we note, many proposals providing automated bandwidth selection strategies are also offered in the statistical literature. Common recommendations for an appropriate criterion to optimise are briefly outlined below and, with respect to these, the prevailing classes of methods for automated bandwidth selection introduced.

3.2.7.1 *Optimisation criteria*

The most popular criteria to measure the performance and accuracy of a density estimate are the Integrated Squared Error (ISE) and the Mean Integrated Squared Error (MISE). These attempt to quantify the difference between the true density and a given estimate. As the name suggests, the MISE is the expected value of the integrated L_2 distance between the density estimate and the true density function f (the ISE). Since the ISE can be treated as a random variable that depends on the true function $f(x)$, the estimator $\hat{f}(x)$, and the particular random sample that is used to obtain the estimate, it is common practice and appropriate to look at the expected value of the ISE, the *mean integrated squared error*. The MISE takes the mean value of the integral to serve as a measure of error between the true function and the estimate of the function.

In the multivariate case these minimisation criterion are formally defined as follows:

$$\mathrm{ISE}(\mathbf{H}) = \int \left[\hat{f}_{\mathbf{H}}(x) - f(x)\right]^2 \mathrm{d}x \tag{6}$$

$$\mathrm{MISE}(\mathbf{H}) = E \int \left[\hat{f}_{\mathbf{H}}(x) - f(x)\right]^2 \mathrm{d}x \tag{7}$$

These criteria obviously coincide asymptotically but for finite samples the kernel bandwidth $\mathbf{H}$ that minimises the ISE and MISE may differ. Both metrics (Equations (6),(7)) make use of an L_2 norm and unqualified integrals refer here (and in all subsequent instances) to integration over the real line $\mathbb{R}$ or whole space. ISE and MISE remain the

most commonly used metrics due to their tractability and wide spread implementation in bandwidth selection software. Some authors also consider KL divergence, Hellinger distance and L_1 metrics (*e.g.* [71]), in attempts to handle cases where L_2 metrics are not appropriate (*e.g.* robustness to outliers) and report appealing properties such as ease of error visualisation. A comprehensive comparison of these distance considerations is found in [70]. Once a metric is selected, the optimal bandwidth is obtained by minimising $\mathbf{H}$ over the space of symmetric, positive definite $d \times d$ matrices. In the cases of ISE and MISE this gives:

$$\mathbf{H}_{\text{ISE}} = \arg\min_{\mathbf{H}} \text{ISE}(\mathbf{H})$$

$$\mathbf{H}_{\text{MISE}} = \arg\min_{\mathbf{H}} \text{MISE}(\mathbf{H})$$

The remaining problem is that the true density function f is generally unknown so these criteria do not result in closed-form expressions. In such cases, the bandwidth selection task becomes that of minimising an approximation to the chosen criteria. At the core of most popular methods involves applying a minimisation strategy to an approximation of the ISE or MISE. There is no clear consensus on which criterion should be chosen due to the fact that no single procedure can be considered optimal in every situation. Some further detail on this debate is given in [266, 160].

3.2.7.2 *Bandwidth selection methods*

Prominent classes of approach that perform optimal bandwidth criteria minimisation tasks can be distinguish between using the optimality criteria outlined previously. The main classes are: *Plug-in* methods (that typically try to minimise the MISE), *rules-of-thumb*, *cross-validation* methods (that consider the ISE) and variable bandwidth estimators. Here these approaches are briefly summarised in relation to the bandwidth selection approach made use of in this work. The interested reader is referred to [122] for an in-depth review.

Plug-in methods:

Plug-in methods tackle the problem of approximating the MISE minimisation criteria by using a Taylor series to construct an asymptotic expression, the AMISE (Asymptotic

MISE), which is then utilised to create tractable bandwidth selectors. Performance often depends on the choices of pilot bandwidths in practice (intermediate bandwidths selected in order to approximate the AMISE, defined in terms of higher order derivatives of the unknown true density f). If good pilot estimators (several have been proposed *e.g.* [49]) are employed then good minima (bandwidth choice) have been reported. Due to this dependency, *plug-in* methods are not entirely data adaptive as they require pilot bandwidth information to make derivative estimates and may, therefore, perform poorly for small sample sizes.

In the case of bivariate data, various works have considered *plug-in* algorithms [272, 79] for obtaining suitable density estimates and have shown that the bandwidth found using this strategy converges in probability to the true optimal bandwidth h_{AMISE}. These algorithms cannot however be directly extended to the general multivariate setting. The underlying principle is that an expression involving an unknown term can be tackled by replacing the unknown term with an estimate. A comprehensive review of *plug-in* methods is provided by [266].

Rule-of-thumb methods:

Simple *rule-of-thumb* methods attempt to optimise the same criteria as *plug-in* methods introduced above and essentially provide a simplified *plug-in* bandwidth selector. Silverman's *rule-of-thumb* [240] is probably the most popular of these. Recall that *plug-in* methods make use of further bandwidth estimators to approximate higher derivatives of f. Silverman's *rule-of-thumb* involves simply estimating f'' directly using a parametric normal density. This reference distribution is rescaled to have variance equal to the sample variance. The approach was originally put forward in [69], where it was proposed for histograms. This procedure provides a good estimate of the optimal bandwidth if the true density function is nearly normal. However, if this is not the case (*e.g.* multimodal densities) Silverman's *rule-of-thumb* is likely to fail. The *plug-in* approach, introduced previously, can be considered a refinement to this *rule-of-thumb* approach.

Cross-validation methods:

Cross-validation bandwidth selectors are the main alternative to *plug-in* selectors and

typically attempt to minimise the ISE. These selectors provide a commonly implemented heuristic for selecting kernel bandwidths and are able to find a data-driven solution without making assumptions about the shape of $f(x)$ or the family of distributions to which the unknown density belongs. Comparing cross-validation with *plug-in* methods, the ISE is considered by some an unrealistic target to minimise as it takes the true density of f into account too much. Methods minimising the ISE can only hope to obtain good results when the sample at hand is "typical" and reflects the structure of the true distribution well. This observation often leads to the counter-claim that it is only reasonable to measure the performance of ISE methods in terms of estimating f in the average case. Furthermore, cross-validation is said to have stability issues for large data sets [237] and "*often under-smooths in practice, in that it leads to spurious bumpiness in the underlying density*" ([241] pp. 76).

A positive point to note is that cross-validation methods allow the selected bandwidth to automatically adapt to the smoothness of f. This is in contrast to *plug-in* methods and Silverman's rule-of-thumb which are less volatile but not entirely data adaptive and may therefore not work well for small sample sizes. Plug-in methods often exhibit faster convergence rates than cross-validation, however making use of the AMISE depends on asymptotic arguments that arguably have less intuitive interpret-ability than the MISE (and ISE).

3.2.7.3 *Variable bandwidth selection*

Common criticisms of the previously surveyed automatic selectors are that cross-validation tends to under-smooth and suffers from high sample variability while plug-in estimates deliver a more stable estimate but typically over-smooth. A fixed bandwidth, found by either approach, may mean that in regions of low density all samples will fall in the tails of a kernel and result in very low weighting, while regions of high density will find an excessive number of samples in the central region producing very high weighting. As is often noted (*e.g.* [230]), increased smoothing is typically required to counter excessive variation in the tails of a distribution where data are scarce while less smoothing is needed near the mode(s) of a distribution to prevent features from being diminished in the resulting estimate. Several qualities typically present in our point set registration task (estimates are multimodal and multivariate) result in a bias-variance trade-off

that drives most global bandwidth choices to estimates that may lack visual appeal and make feature recognition difficult.

Such situations have motivated the notion of variable bandwidth functions that allow varying amounts of smoothing depending on local characteristics of the data and the density being estimated. Introducing a variable bandwidth attempts to fix the highlighted problems by varying the *width* (and *shape*) of a kernel in different regions of the sample space. Allowing the bandwidth to vary provides the flexibility to use smaller bandwidths (and reduce the bias) in regions where there are many observations, and larger bandwidths (reducing the variance) in regions where there are relatively few observations. This freedom makes variable bandwidth estimation a particularly effective technique when the sample space is multi-dimensional [39]. The term *variable kernel estimates* was introduced in [32] which took multivariate densities into consideration and investigated a local bandwidth such that kernels each have their own size and orientation regardless of where the density is to be estimated. In [32] it is originally suggested that using a local bandwidth such that $h(x_i)$ is the distance from x_i to the k-th nearest data point which remains a popular strategy and also inspires the adaptive bandwidth selection strategy introduced in this thesis (see section 3.3.4.1 for further detail).

Further approaches set an individual bandwidth h_i for each query point by utilising a *pilot* density estimate (*i.e.* an initial fixed bandwidth kernel estimate of the density, *c.f.* section 3.2.7.2). In this manner the work of [3] select each h_i to be inversely proportional to the square root of the density at x_i by making use of a *pilot* estimate to obtain an initial estimate for $f(x_i)$. It is noted by [240] that this method of producing an initial density estimate is insensitive to the fine detail of the chosen *pilot* (commonly Gaussian).

Since this initial variable bandwidth work, two main strategies of selection have evolved. Rather than using a single bandwidth matrix $\mathbf{H}$ to estimate f at every query point x, the first strategy employs a bandwidth matrix $\mathbf{H}(x)$ that varies according to the *query point* x at which an estimate of f is required. This is referred to as a *balloon* estimator and takes the form:

$$\hat{f}_{\mathbf{H}(x)}(x) = \frac{1}{N}\sum_{i=1}^{N} K_{\mathbf{H}(x)}(x - p_i)$$

The *balloon* estimator was first introduced in the form of the k-th nearest-neighbour estimator. In [161], $\mathbf{H}(x)$ was based on a suitable k such that the bandwidth was a measure of distance between x and the k-th data point nearest to x. In this way the kernel width is varied to make it proportional to the density at the query point.

A second variable bandwidth strategy involves having the bandwidth $\mathbf{H}(A_j)$ vary with the set of *observed* data points A_j in some neighbourhood of the query point x_j. This type of estimator is often known as a sample-point or point-wise estimator and an initial example of this strategy is attributed to [32]. Analogously it takes the form:

$$\hat{f}_{\mathbf{H}(A_j)}(x_j) = \frac{1}{N}\sum_{i=1}^{N} K_{\mathbf{H}(A_j)}(x_j - p_i)$$

The introduced strategies for density estimators have been studied extensively. Jones *et al.* [140] give a comparison of such estimators in the univariate case while Terrell, Scott and Sain [256, 225] have examined both formulations in the multivariate setting. Applying variable bandwidth techniques to computer vision problems remains a popular approach and includes recent work on *e.g.* background subtraction, blob detection and hand-written digit recognition, amongst others [62, 185, 255].

In this work we introduce a hybrid balloon estimator using a nearest-neighbour approach such that the size and shape of each kernel is affected by sample points in the neighbourhood of the query point. We provide detail of this bandwidth selection approach in section 3.3.4.2.

3.2.7.4 *Discussion*

It is clear that there is not a single procedure to determine the optimal bandwidth in every problem instance. The optimal method in each case depends upon both the available samples and the particular goal of the density estimate. Many automatic methods make strong assumptions that go against core ideas of non-parametric density estimation. Experimentation is nearly always required, as different kernel widths and shapes may provide different information about the data. Moreover, even bandwidths selected using asymptotically optimal criteria may show poor behaviour in simulation ([119]). As a consequence, one valid approach is to determine bandwidths by different selection methods and compare the resulting density estimates. The practitioner is generally faced with a formidable computational cost for appreciable data set sizes and

[112] note that this becomes even more prohibitive when models with different kernel bandwidths must be evaluated to find an optimal model.

Kernel bandwidth values should be influenced by the purpose for which the density is to be used. This in turn makes the purpose of the density estimate an influential factor in choosing a bandwidth selection strategy. For example, a good density for estimating an unknown curve is not necessarily also good for prediction tasks [240, 118]. Nevertheless, an automatically selected bandwidth (*e.g.* using the surveyed methods) is often a good starting point. In the following section, we introduce the kernel and bandwidth selection choices made use of in this work and justify these decisions in relation to the problem domain addressed and the techniques surveyed previously. Further detail on bandwidth selection can be found in [118, 195, 275].

3.3 DENSITY ESTIMATION FOR POINT SET REGISTRATION

3.3.1 *Point set registration*

Point set representations regularly emerge in a diverse array of applications for computer vision, computer graphics, medical image analysis and reverse engineering. Many challenging problems in these fields can be addressed by making use of input data formulated as, or summarised by, point sets. We focus on the important problem of point set registration, which is encountered in areas such as stereo correspondence, shape matching, feature-based image registration and model-based segmentation.

Point set registration can also be considered one of the crucial stages of surface modelling and surface reconstruction from range data. The ability to easily create three-dimensional models of physical entities and environments from depth data finds useful applications in many of the fields highlighted above. Prominent examples include autonomous navigation [278], accelerating the production of special effects and computer games [289], various medical imaging applications (*e.g.* [60], [177]) and preserving cultural heritage [133].

A rich history of work exists on registering *pairs* of point sets (see Chapter 2 for review) that has resulted in fast and reliable algorithms for the task. The goal of reconstructing models of scenes and objects from range data has facilitated a natural progression to the study of the *multi-view* registration problem which has now also

gained significant attention in the vision and graphics communities. The process of estimating transforms between the point sets and generating complete object representations by fusing information from the partial views into a common coordinate frame is known as the *multi-view* registration problem.

Here a statistical method to perform the multi-view registration of point sets is proposed. Multiple object (or environment) viewpoints can be generated by varying depth sensor (or target) position. Viewpoint depth information is then represented by point sets typically in the form of 3D point clouds. The proposed method uses a non-parametric kernel density estimation scheme. Kernel density estimation is a fundamental data smoothing technique where inferences about a population are made based on finite data samples (density estimation principles are covered in sections 3.2.1 - 3.2.7). We define a density function that reflects the likelihood that a point $\mathbf{x} \in \mathbb{R}^3$ lies on the unknown true surface $\mathcal{S}$ which is observed by point samples $\mathcal{P}$. This surface estimate is then used to guide view registration in the sensor transform space as we alternatively refine view pose positions and our model surface estimate. Many algorithms have incorporated an update scheme wherein transforms and correspondences are alternatively optimised while keeping the other fixed [21], [64], [90], [276] and [57]. By alternating the update of the transforms and correspondence parameters, the two solutions tend to mutually improve one another during the process and converge to a reasonable (albeit possibly sub-optimal) solution. Data sources and representations made use of in our point registration work are briefly introduced and the multi-view registration problem is discussed before going on to formally define our particular density estimation contributions relating to 3D point cloud data.

3.3.2 *Data sources and representations*

The Ohio State University range image database [192] is a popular collection of range images made use of in our experimental registration work. The database contains images of various objects with depth data available in greyscale GIF and sets of x,y,z fixed-point measurements that can be used to produce point clouds. The range images are obtained from both structured-light range sensors (courtesy of Michigan University) and from Ohio State University's Minolta 700 range scanner. In addition to the data sets obtained from real range sensors, the database also contains images from synthetic

models. The 3D models used to synthesise imagery are also available as part of the wider OSU 3D database [192]. Example range images of sample objects from this resource are provided in Figure 9.

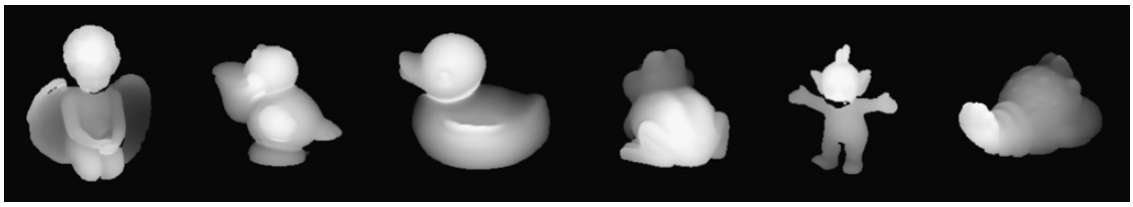


Figure 9: Example range images of 6 data sets from the OSU range image database [192].

Further synthetic data sets are created by generating point clouds from simple mathematical functions and primitive geometrical shapes (see section 3.5 for details). Additionally, point cloud data sets are obtained from physical objects locally by making use of stereo camera systems. These point clouds are derived, using a standard pinhole camera model, from depth maps obtained using propriety stereo correspondence software [73]. The Microsoft Kinect sensor [183] is also used to capture depth maps that are made use of in Chapter 5. Further details on locally captured data sets and experimental work carried out using them are found in Chapter 5.

A point cloud in n-dimensional space can be defined as a set of N points $\mathcal{P} = \{p_i \in \mathbb{R}^n \mid i = 1, \ldots, N\}$. Our experimental work mainly concerns registering point clouds representing 3D spatial measurements, obtained from depth sensors, thus point sets experimented with typically constitute sets of triples $\{(x, y, z) \mid \forall x, y, z \in \mathbb{R}^3\}$. In summary the experimental work in this chapter makes use of a variety of depth sensor measurement data sets, represented by 3D point clouds, providing varied input data in an effort to explore and challenge our point set registration framework.

3.3.3 *Multi-view registration*

An initial coarse alignment of multiple viewpoints can often be found directly from the sensor scanning system or interactively provided by the user. Modern depth sensors, capable of capturing many frames per second, often provide a natural coarse alignment as the spatial transforms between consecutive views are likely to be small. Using such a coarse alignment as input, spatial registration can then be refined by accurately registering the overlapping parts of the viewpoints. This refined registration task is typically

subdivided into the correspondence and alignment sub-problems. The correspondence problem is defined as: given a point in one scan, determine the samples in other viewpoints that represent the same physical point on an object surface. Note that with data measurements from physical surfaces, an exact correspondence may not actually be sampled due to sensor quantisation. The alignment problem involves estimating the motion parameters that bring one scan into the best possible registration with the others. Providing a ground truth for either of these objectives renders the other trivial to solve.

As discussed in Chapter 2 (section 2.2), a simple method for accurately registering many viewpoints involves *sequential* registration. The highlighted disadvantages of this method included error accumulation, propagation and the necessary property that view sequence order must be known or manually specified due to non-zero view overlap constraints. In this work, we consider an alternative multi-view registration approach of *simultaneous* global registration, where the aim is to align all views simultaneously by distributing registration errors evenly between overlapping viewpoints. Previous techniques that fall into this category of approach for tackling the multi-view problem are surveyed in Chapter 2 (section 2.2).

Specifically, to register multi-view point cloud data from the set of views $\{V_1, V_2, \ldots, V_M\}$ we firstly infer the likely true underlying surface structures from the potentially noisy, coarsely aligned set of views. For each view V_m we wish to register, we use kernel density estimation to construct a surface approximation S_m that takes into account the current position of all other viewpoints $\{V_n | n = 1, \ldots, M \wedge n \neq m\}$. We use this inferred surface to optimise the spatial pose of V_m, transforming the view in pose space and assessing updated poses by creating and evaluating an energy function defined in terms of how well the moving view V_m is aligned with the view set surface approximation S_m (defined by the density estimate).

This process is performed for each viewpoint V_m that we wish to register *simultaneously*. By updating the pose of all views simultaneously to positions of high energy (collective high density) we effectively move each view to a best fit position that maximises the likelihood that the view pose of V_m concurs with the current corresponding surface estimate S_m (and therefore implicitly with other optimised scan positions). In the following sections (3.3.4 - 3.3.5) formal detail is provided on the kernel density approach and energy functions we define and optimise to improve the registration of view

V_m to inferred surface S_m. Detail on the multi-view alignment aspects of the strategy are then provided in section 3.4.

3.3.4 *Density estimation for 3D point clouds*

Here we outline our density estimation approach that provides object surface estimates from depth data and how we utilise these estimates for point set registration. The method we introduce for estimating surfaces can be considered a non-parametric density estimation scheme. Given many partial surface views in the form of sets of depth samples, we estimate a kernel density function of the data to determine a point-based approximation of the sampled surface. We use this density to guide an energy minimisation in the transform space, aligning all partial views robustly. Here robustness implies that a surface estimation is able to cope with noisy data that may contain a small fraction of gross measurement errors. A concise introduction into the field of robust filtering and estimation is available in [248]. In this way, the registration technique that we develop is capable of handling noisy sets of points, sampled from object surfaces, that may contain measurement noise and other outliers (see Chapter 5, section 5.4.2.1 for experimental evidence supporting this claim).

By analysing measurement uncertainty and variability in point-sampled geometry we build a representation that focuses on using discrete surface data stemming from 3D acquisition devices where a finite number of (possibly noisy) samples provides information about an underlying unknown physical surface. We attempt to capture this measurement uncertainty by introducing a statistical representation that quantifies, for each point in space, the plausibility that the point is in a well registered spatial position in relation to an implicit surface that fits the available data. This produces a statistically likely generating surface in accordance with measurements offered from each viewpoint. Our estimate is an adaptation of the generic kernel density estimation technique outlined previously in section 3.2.6. The six standard steps of a pairwise registration process [223] note an explicit outlier removal stage. The strategy we propose here will implicitly assign low weight to outliers to the point of effective exclusion from consideration and we therefore do not specify an explicit outlier removal step. We use local density maxima to guide our estimate of where the sampled surface is most likely to exist and in turn update scan positions in relation to this inferred surface by spatial

parameter optimisation in the transform space. Here we first discuss kernel and bandwidth properties and then move on to multi-view and transform space optimisation aspects of the approach.

3.3.4.1 *Kernels for 3D point cloud density estimates*

We implement a kernel function with properties that can be considered suited to the nature of the multivariate spatial depth data considered in this problem domain. Similar density kernel components have previously been shown to work well with point cloud data for *e.g.* noise cleaning tasks [228]. The first component (of two) that the kernel, centred on sampled data point $\mathbf{p}_i$, contributes to the energy function evaluated at point $\mathbf{x}$ involves a local plane fitted to a spatial neighbourhood of $\mathbf{p}_i$. This plane is fitted using all points located within a spatial distance h (the bandwidth) of $\mathbf{p}_i$ (see Figure 10 for a 2D example fit and section 3.3.4.2 for further details on bandwidth selection).

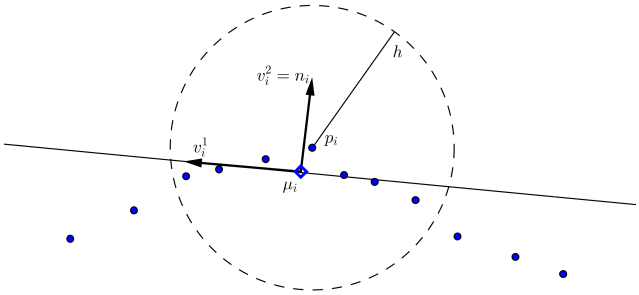


Figure 10: Two dimensional example of our projective distance kernel-component construction. For the kernel centred on data point $\mathbf{p}_i$ we find a least-squares line (plane with trivariate data) fit through the neighbouring sample points (blue) where neighbouring points considered are defined to be within bandwidth distance h. The point μ_i is the centroid of the neighbouring points and eigenvectors v_i^l are found using the local point set covariance.

In practice we fit a least-squares plane (normal n_i, centroid μ_i) to the points $\mathbf{p}_j$ in the spatial neighbourhood of $\mathbf{p}_i$ as dictated by the bandwidth distance h. A method detailing appropriate selection of values for h is provided in the following section 3.3.4.2. The centroid μ_i is the weighted mean of the spatial neighbours of $\mathbf{p}_i$ and the plane

normal n_i is found by applying singular value decomposition to a weighted covariance matrix Σ_i such that neighbouring points $\mathbf{p}_j$ nearer to $\mathbf{p}_i$ are given higher weighting:

$$\Sigma_i = \sum_{\mathbf{p}_j \in \text{Neighb}(\mathbf{p}_i)} (\mathbf{p}_j - \mu_i)(\mathbf{p}_j - \mu_i)^T \chi(\mathbf{p}_j, \mathbf{p}_i) \tag{8}$$

where

$$\chi(\mathbf{p}_j, \mathbf{p}_i) = \frac{1}{\sqrt{\left(\mathbf{p}_j^x - \mathbf{p}_i^x\right)^2 + \left(\mathbf{p}_j^y - \mathbf{p}_i^y\right)^2 + \left(\mathbf{p}_j^z - \mathbf{p}_i^z\right)^2}}$$

We choose a simple reciprocal Euclidean distance weighting for χ, providing a monotonically decreasing weight function based on spatial distance. Since Σ_i is symmetric and positive semi-definite the eigenvalues λ_i^l, $l = 1, 2, 3$, are real-valued and non-negative such that: $0 \leq \lambda_i^3 \leq \lambda_i^2 \leq \lambda_i^1$ and the inverse covariance matrix Σ_i^{-1} can be used to define an ellipsoid G_i with centre μ_i:

$$G_i = \{\mathbf{x} \mid (\mathbf{x} - \mu_i)^T \Sigma_i^{-1} (\mathbf{x} - \mu_i) \leq 1\} \tag{9}$$

where the least-squares fitting plane is spanned by the two main principal axes v_i^1,v_i^2 forming an orthonormal basis and the third v_i^3 provides the plane normal n_i that we require. A schematic example of this is depicted in Figure 11. If normals are provided by the scanning device we can use them instead of the fitted estimates.

The distance from spatial point $\mathbf{x} \in V_m$ to this local fitted plane determines the value of the first component (of two) that the local energy $K_{m,i}(\mathbf{x})$ contributes. Object surface structure can be considered locally planar for sufficiently close proximity and measurement points in well registered positions will therefore lie on or near these locally planar regions. We orthogonally project $\mathbf{x}$ onto the plane and using the squared distance, $[(\mathbf{x} - \mu_i) \cdot n_i]^2$, we measure the first term of the local contribution $K_{m,i}(\mathbf{x})$ as:

$$[h^2 - [(\mathbf{x} - \mu_i) \cdot n_i]^2]$$

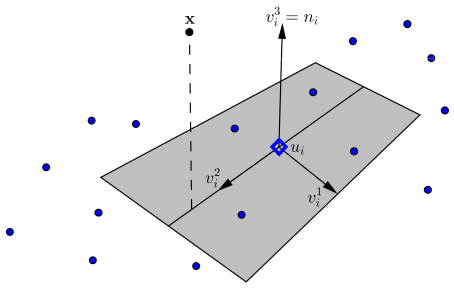


Figure 11: In the 3D case we orthogonally project $\mathbf{x}$ (our density query point) to the locally fitted neighbourhood plane and find the energy contribution of $K_{m,i}(\mathbf{x})$ using n_i to provide our estimated plane normal and μ_i as our weighted neighbourhood centroid. The distance from $\mathbf{x}$ to this fitted plane (dashed line) dictates the contribution of the local energy for the plane fit related to sample $\mathbf{p}_i$.

The bandwidth h provides the maximal distance that points may lie from $\mathbf{p}_i$ and still contribute to the estimation of the local plane that we project the query point $\mathbf{x}$ to. The value of this first $K_{m,i}(\mathbf{x})$ term is therefore greater than or equal to zero by definition and positions $\mathbf{x}$ closer to our locally fitted surface structure are assigned higher energy than positions that are more distant. We claim that this orthogonal plane projection term is a useful kernel component as it provides us with a good measure of point registration error. It can be observed that for sufficiently close proximity, our 3D surface structure can be considered locally planar. Query points in well registered positions will therefore lie on or near these locally planar regions.

Like many common kernels, an additional assumption is that the influence of point $\mathbf{p}_i$ on the estimated density at position $\mathbf{x}$ diminishes with increasing distance. To account for this fact we make use of monotonically decreasing weight functions ϕ_i to reduce influence as distance increases. Our second kernel-component therefore follows [228] and makes use of a trivariate anisotropic Gaussian function ϕ_i, that we adapt to the shape of ellipsoid G_i (Equation 9). This provides the additional property that the distance weighting component is adapted to the point distribution in the spatial neighbourhood of $\mathbf{p}_i$. This allows the kernel *shape* to adapt to the local point distribution. In practice we estimate $\phi_i(\cdot)$ parameters μ_i, Σ_i by reusing the same neighbouring points of $\mathbf{p}_i$ according to the bandwidth distance h. From these points we reuse the neighbourhood mean vector μ_i and weighted covariance matrix Σ_i (Equation (8)). Making use of Σ_i again in this second term provides an anisotropic weight derived from neighbouring

points such that their distance from $\mathbf{p}_i$ dictates their influence on the shape of the kernel component. In summary, the second contribution to the local kernel is a trivariate Gaussian weighting:

$$\phi_i(\mathbf{y}) = \frac{1}{(2\pi)^{3/2}|\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{y}-\mu_i)^T \Sigma_i^{-1}(\mathbf{y}-\mu_i)\right)$$

The product of the local projective plane distance term and this trivariate Gaussian term provide the local kernel contribution, centred on neighbouring point $\mathbf{p}_i$, to the energy function evaluation of point $\mathbf{x}$:

$$K_{m,i}(\mathbf{x}) = \phi_i(\mathbf{x}-\mu_i)^{\alpha} \cdot \left[h^2 - [(\mathbf{x}-\mu_i)\cdot n_i]^2\right]^{(1-\alpha)} \tag{10}$$

The points $\mathbf{x}$ that we evaluate are spatial samples belonging to view V_m and α provides a tuning parameter that allows the influence of either kernel component to be amplified or diminished (see section 3.5 for further detail). This leaves us to define the full energy function $\hat{E}_m(\cdot)$ modelling the likelihood that a point $\mathbf{x}$ is currently lying on the unknown true surface approximated using points in the set of views $\{V_n | n = 1,\ldots,M \wedge n \neq m\}$. This involves accumulating and summing the local $K_{m,i}(\mathbf{x})$ contributed by all points $\mathbf{p}_i$ in the spatial neighbourhood of $\mathbf{x}$ as defined by the bandwidth h:

$$\hat{E}_m(\mathbf{x}) = \sum_{\mathbf{p}_i \in \text{Neighb}(\mathbf{x})} w_i K_{m,i}(\mathbf{x}) \tag{11}$$

We are able to incorporate scanning confidence measures $w_i \in [0,1]$ associated with each measurement point $\mathbf{p_i}$ by scaling the amplitudes of our energy functions. If no scanning confidences are provided we use $w_i = 1\ ,\ \forall i$.

The main motivation for formulating an energy function based on density estimation to infer surfaces is our desire to solve the multi-view registration problem. The kernel method provides a means to infer where physical surfaces exist that (1) improve in confidence with additional data, (2) has a natural ability to account for outliers and view misalignment and (3) provide smooth gradients for an (arbitrary) iterative optimisation process, we provide strong justification for our strategy choice. Further, only a

limited, finite number of sensor points are available to represent underlying continuous object surfaces and additionally points are typically obtained from a depth sensor that potentially produces noisy measurements. The location of every point is, therefore, partially uncertain and we make use of density estimation tools in an attempt to alleviate noted negative effects [137] that measurement noise can have on the quality of point registration.

Kernel density estimation requires the entire set of data samples to be stored to produce a density estimate. This has merit in that there is no computation involved in a model "training" phase because this simply requires storage of the data set. However, this is also commonly noted to be one of the major weaknesses of the approach [24] because the computational cost of evaluating the density grows linearly with the size of the data set. This can often lead to expensive computation if the data set is large, such as is often the case with the application considered here (sets of spatial point measurements over many viewpoints). This effect can be partially offset, at the expense of some additional one-off computation. Constructing tree-based search structures allows nearest-neighbour points to be found efficiently during kernel construction, avoiding the need to perform exhaustive distance searches on the data set. In practice we make use of k-d tree structures [17] for this purpose. See Chapter 4 for further options explored to mitigate computational cost.

3.3.4.2 *Adaptive bandwidths for estimating point cloud density*

As noted in section 3.2.7 one of the difficulties with the standard kernel approach to density estimation is that the bandwidth parameter h, dictating kernel width, is often fixed for all kernels. In regions of high data density, a large value of h may lead to over-smoothing and a washing out of structure that might otherwise be extracted from the data. However, reducing h may lead to noisy estimates elsewhere in the data space where the density is smaller [24]. Thus the optimal choice for h may be dependent on the location within the data space. The standard technique for addressing this problem involves adaptively defining a unique bandwidth for each kernel (see section 3.2.7.3 for common approaches).

Multi-view registration tasks commonly contain data sets that exhibit varying levels of measurement redundancy in surface sampling locations and therefore distinct physical areas may be sampled at varying densities. In this problem domain, washing out of

structure tends to manifest as over-smoothing of distinctive surface features and detail that might prove useful during the registration process. Alternatively reducing a bandwidth too much can result in fitting (and fabricating) unwanted surface structure to small outlying depth measurements caused by *e.g.* sensor noise. Additionally, constant kernel bandwidths may not be suitable for view sets with coarse initial alignment or high sensor noise. For these reasons adaptive kernels are explored in this work as part of the multi-view point cloud registration process. Here adaptive kernels are instantiated using *balloon*-like estimators (see section 3.2.7.3) that make use of nearest-neighbouring data samples. The k-nearest-neighbour kernel density estimate, originally proposed in [166], is given by:

$$\hat{f}_{h(\mathrm{KNN}(\mathbf{x}))}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} K_{h(\mathrm{KNN}(\mathbf{x}))}(\mathbf{x} - p_i) \tag{12}$$

where $h(\mathrm{KNN}(\mathbf{x}))$ provides a kernel bandwidth defined as the Euclidean distance between the query point $\mathbf{x}$ and the k-th nearest-neighbour of point $\mathbf{x}$ among the available point samples:

$$h(\mathrm{KNN}(\mathbf{x})) = \min_k \left(\{ |\mathbf{x} - \mathbf{p}_i| \;\middle|\; \mathbf{p}_i \in \mathcal{P} \} \right)$$

where $\min_k(\{d\})$ is the k-th smallest member of the set $\{d\}$. In the case of multi-view registration; sample $\mathbf{x} \in V_m$ and we find $h(\mathrm{KNN}(\mathbf{x}))$ by considering the Euclidean distance to members of amalgamated point set $\mathcal{P}$ where $\mathcal{P}$ is the union of all points belonging to viewpoints $\{V_n | n = 1, \ldots, M \wedge n \neq m\}$. This KNN Euclidean distance, that varies with sample location, is the bandwidth value assigned to h in the bi-component kernel $K_{m,i}$ centred on each point p_i (Equation 10) that contributes to the energy evaluation at point $\mathbf{x}$. Using this approach, the distance to the k-th neighbour now governs the degree of density smoothing and again there is an optimal choice for k that is neither too large nor too small. We concede that this introduces a new parameter that must be determined however, in comparison to a globally fixed value for h, this approach inherently allows for adaptive behaviour in the local spatial distribution of data samples. We note that while density estimation using an optimal fixed global bandwidth, obtained using *e.g.* AMISE based techniques (see section 3.2.7.2), allows the density estimate to converge in probability to the true density f, the integral of a KNN density

estimate is usually very close to 1, but is not exactly 1 [193]. This implies that the density estimate produced using k-nearest-neighbour kernels is not a *true* density model because the integral over the entire data space diverges [24]. However, in practice we find this method of bandwidth selection advantageous in conjunction with our kernel construction as the *shape* of our local kernels is varied with the Gaussian component (the shape of the Gaussian is fitted to the local point k-neighbourhood) and we vary the *size* of each kernel by defining the projective distance component in terms of the distance to the furthest nearby neighbouring observation.

Motivation for this adaptive bandwidth selection strategy can be observed in Figure 12. If a small, fixed bandwidth h is used to construct density estimates, local maxima of $\hat{E}_m(\cdot)$ can be observed distant from the most likely surface in regions of misaligned point clouds and large-amplitude noise. During transform optimisation these maxima may in turn attract data to an erroneous alignment in the registration process. The alternative of adapting kernel sizes locally by varying h in relation to local density and requiring a k-neighbourhood contribution to each density estimate leads to larger kernel sizes in regions of misregistration and large-amplitude noise due to the typically lower sampling density. The fixed bandwidth density estimation in Figure 12c illustrates such local maxima.

The globally fixed spatial distance h results in the density estimate at *some* query points (Figure 12c) being defined by as many as 25 local kernels yet, in sparser regions, as few as 2 sample kernels are near enough to take part in local summations at density query points. Alternatively in Figure 12d we force KNN=25 for kernel building and therefore the 25 spatially nearest kernels, attributed to the 25 nearest data samples, contribute to the density estimate at *each* and *every* query location $\mathbf{x}$. This allows density estimate locations, with values defined by inconsistently distributed neighbourhood samples, to adapt spatially. This in turn helps to dampen the effects of local maxima and sensor noise. This typically results in smoother and more stable surface estimation in areas where samples contain large scale measurement noise or view misalignment.

Additionally, scan misalignment has the potential to form "view cliques" during the registration process, creating regions of unwanted multi-modal density. Cliques are found when sets of scan views form groups such that views *within* a clique are well registered, but the cliques themselves are not well registered to each other. This is a common problem found in previous multi-view registration strategies and is discussed

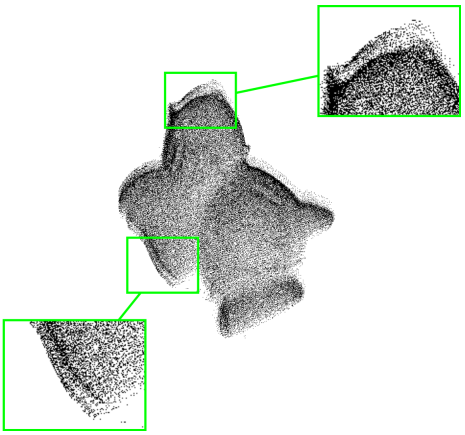
in *e.g.* [83]. Some typical "view clique" misalignment can be observed in Figure 12a (enlarged areas). A simple example consists of a set of scans that form two cliques such that each scan is well aligned within a clique but not between cliques. For point registration techniques that make use of *e.g.* minimising exact point pair matching distances, if each point is always paired with a point member (*e.g.* the nearest point) from within its own clique, the *intra*-clique registration may be satisfying but such a pairing will prevent the *inter*-clique registration from improving.

The alternative approach introduced here, involving registering scans to a surface approximation by way of querying a density estimate, essentially defines a *soft* correspondence between points (see *e.g.* [58] for discussion on previous soft correspondence work). A soft correspondence approach, in conjunction with the introduced adaptive bandwidth selection strategy, is capable of addressing the "view clique" problem by selecting appropriate bandwidths that result in density estimates (and surface representations) that can merge and consolidate sample regions exhibiting typical "view clique" behaviour. In comparison, global bandwidth strategies may result in unwanted multi-modal estimates in such regions.

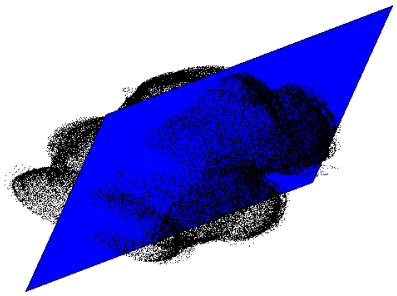
In Figure 12c energy function values are obtained by querying the planar segment slice found in Figure 12b using the OSU "Bird" data set. The coarse alignment configuration of the viewpoints is found in 12a. The energy function provides a surface location estimate. Function values are represented by colours increasing from deep blue to red. Figure 12c exhibits a small, globally fixed spatial bandwidth h. The density estimate at each point draws on local kernel contributions that lie within a spatial distance h. Misaligned viewpoints and sensor noise (zoomed areas) often result in unwanted multimodal maxima of our energy function $\hat{E}_m(\cdot)$, potentially distant from the most likely true surface. Such ragged surface approximations often prevent views being drawn into a better registration. Disparate local maxima prevent agreement on the location of a globally consistent surface.

In comparison, in Figure 12d surface approximation of the data set uses an adaptive k-neighbourhood to define kernel bandwidth locally. The density estimate at each point is required to draw on the k-nearest kernel contributions regardless of spatial distance. Note that our surface approximation becomes a smooth function. Outlying maxima, due to view misalignment and sensor noise, are well damped and diminished. The possibility of view cliques developing during registration is reduced. The density stability in the

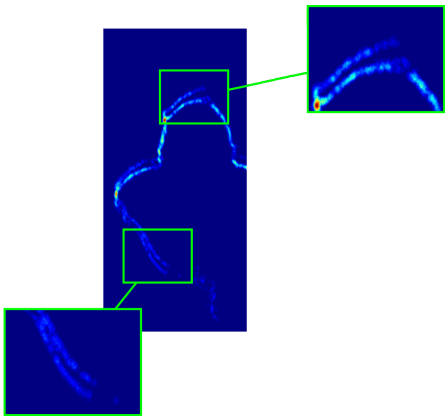
highlighted regions is visually improved and misaligned regions are smoothed to form a consistent surface adaptively.



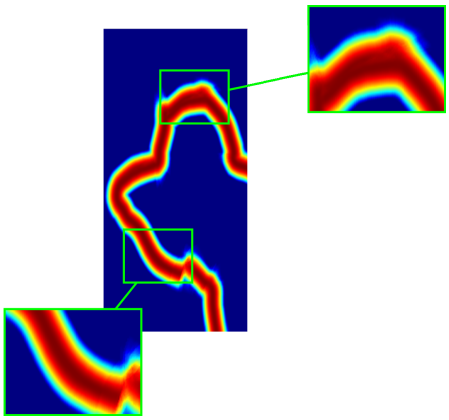
(a) OSU "Bird" data set exhibiting partial scan misalignment (see section 3.5.2 for OSU data details). Note highlighted areas of view misalignment.



(b) A planar slice through the coarsely aligned "Bird" data set where our registration energy function is queried for exposition. See below for zooms of the slice region.



(c) Density estimation utilising a small, globally fixed spatial bandwidth h. See text for detail, best viewed in colour.

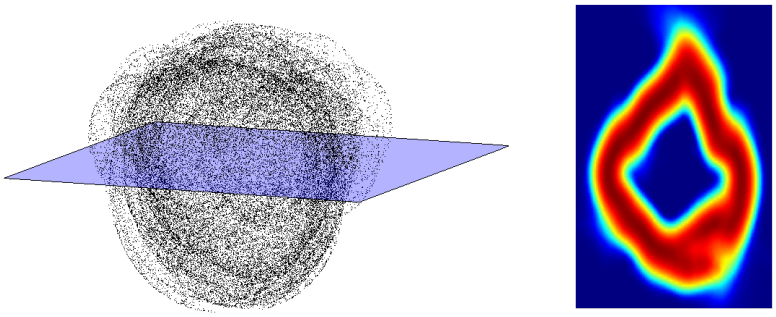


(d) Density estimation using our adaptive k-neighbourhood to define kernel bandwidth locally. See text for detail, best viewed in colour.

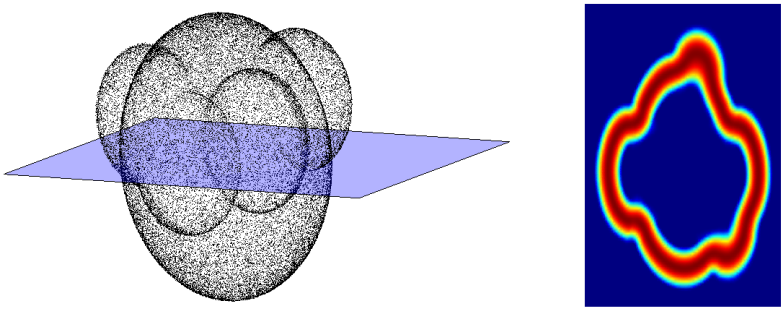
Figure 12: The effects of fixed and adaptive kernel bandwidth choices.

Some further justification for applying an adaptive bandwidth strategy to the registration problem is provided in Figure 13, where density estimation is applied to synthetic point cloud data. Figure 13a (left) shows a synthetic data set containing a collection of 20 individual 3D point sets. Each point set simulates depth scan measurements of an object from a particular point of view. In practice, we first generate a complete syn-

thetic geometrical object surface (in the case of Figure 13, from a collection of simple sphere-like bulbous shapes). Camera/sensor positions are then simulated to generate each scan viewpoint by sampling point measurements from the synthetic surface area visible to the synthesised-camera.



(a) Left: A synthetic data set containing 20 point sets representing partial-view object depth scans in a coarsely perturbed configuration. The planar slice through the data set indicates where we evaluation our energy function for visualisation. Right: A visualisation of $\hat{E}_m(\cdot)$ at points in the planar slice through our synthetic data set.

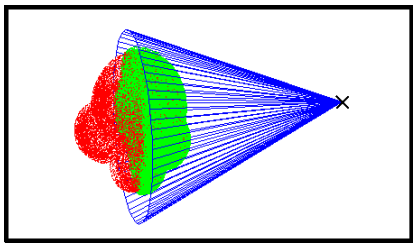


(b) The synthetic data set after ten iterative steps of simultaneous viewpoint pose optimisation in the transform space. The surface approximation is iteratively improved, see text for details.

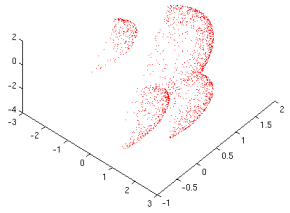
Figure 13: Synthetic point cloud data representing object depth scans and related energy functions evaluated at planar slices.

Simple non-symmetric objects and surface structures are used to prevent degenerative view-registration solutions. Sets of point measurements are created that correspond to the part of the object in the current field of view (see Figure 14 for an example of a single resulting point set). The spatial position of each viewpoint is then collected in the same frame of reference and randomly (rigidly) transformed such that the set of resulting views represent a coarsely perturbed view configuration (see sections 3.5.1.3–

3.5.1.4 for further synthetic dataset construction details). Figure 13a (left) shows this coarsely perturbed collection of viewpoints and a planar slice indicating locations where the energy function is queried for exposition. Figure 13a (right) shows the $\hat{E}_m(\cdot)$ energy function values at the slice region location, represented by colours increasing from deep blue to red.



(a) Simulated camera/sensor position and frustum. Each scan viewpoint is generated by sampling point measurements from the synthetic surface area visible to the synthesised-camera.



(b) A resulting partial point cloud representing an object viewpoint generated from the synthetic sphere-like, bulbous object surface.

Figure 14: Simulated camera/sensor and synthetic point cloud data generation.

The set of scans, Figure 13a (left), exhibit coarse misalignment due to the viewpoint spatial perturbation, however using a KNN adaptive bandwidth density estimate results in an energy function with a smooth nature that in turn aids scan pose parameter search. Figure 13b (left) presents the same data set after iteratively performing simultaneous viewpoint pose optimisation in the transform space where the registration of the set of viewpoints is visually much improved and Figure 13b (right) illustrates how the energy function becomes tighter (and the estimated surface location more confident) due to the iteratively improved viewpoint alignment combined with our adaptive kernel bandwidth. This illustrates the benefits of iterating between optimising the latent surface estimation and optimising the alignment between the estimate and the input partial-view depth scans. See section 3.4 for registration algorithm details and section 3.5.1 for further comparison of synthetic data registration results to ground truth poses.

As all viewpoints are simultaneously aligned and brought into positions of *tighter* registration, the mean distance between depth sample measurement points (μ inter-point distance) decreases. As the sampling density of a region increases, our KNN

adaptive-bandwidth strategy is able to naturally avoid over-smoothing of detail by intrinsically reducing the spatial area used for density estimation at each query point. Figure 15 illustrates how the kernel bandwidth size, defined as the KNN Euclidean distance, evolves during a typical registration process (only the mean value per point cloud plotted for clarity). It can be seen how kernel bandwidth sizes reduce as we iteratively apply spatial transforms to each point cloud and draw them into a tighter alignment. The technique is therefore capable of fitting surface structure to emerging object detail as viewpoints move into positions of better registration. The registration strategy takes advantage of this adaptive bandwidth by iteratively switching between optimising the latent surface shape / location and optimising the alignment of viewpoint sets in relation to this surface. Sections 3.3.5 and 3.4 provide further detail on this registration strategy.

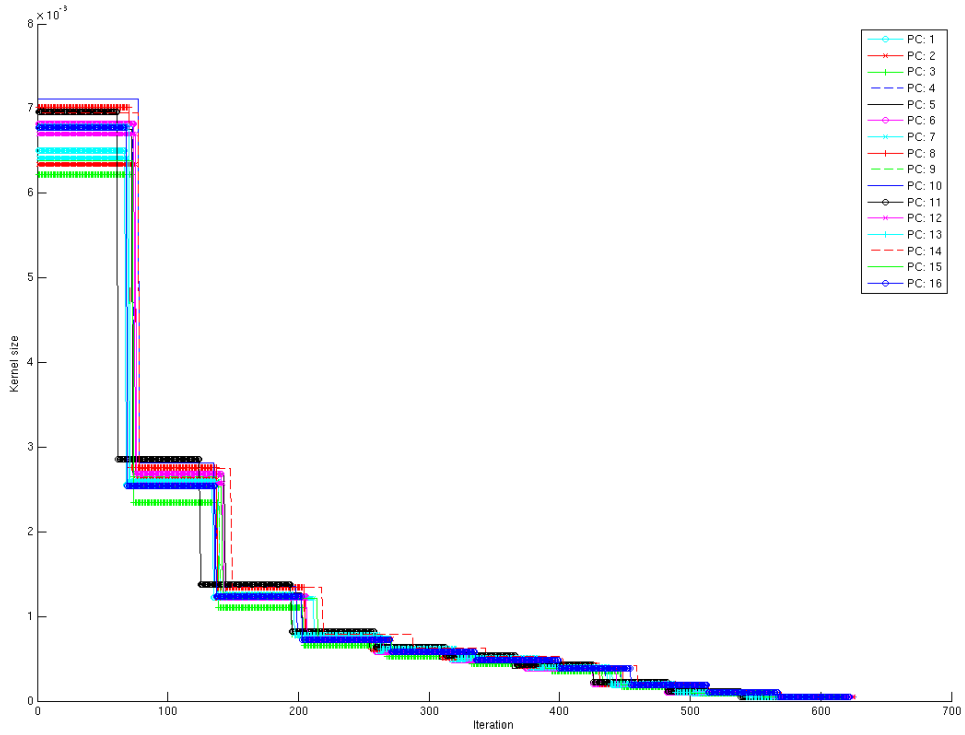


Figure 15: Local kernel adaptive bandwidth size (as defined by KNN Euclidean distance) for each point cloud in a synthetic dataset containing 16 viewpoints. Mean bandwidth values for each viewpoint are plotted versus viewpoint transform registration iterations. Adaptive bandwidths are seen to decrease in size for each viewpoint during registration as views fall into tighter alignment.

3.3.5 *Energy functions for evaluating registration quality*

The core of the registration strategy involves defining a smooth energy function $\hat{E}_m(\cdot)$ (Equation 11) that reflects the likelihood that a point $\mathbf{x} \in \mathbb{R}^3$ is a point spatially near the inferred surface S_m, where S_m is estimated using the current alignment of partial views $\{V_n | n = 1, \ldots, M \wedge n \neq m\}$ and the points $\mathbf{x}$ we are interested in querying are spatial measurement samples belonging to view V_m. Using this estimate of the underlying surface we are able to guide view registration by way of optimisation in the transform space. Once view positions have been simultaneously and independently optimised we can iteratively re-estimate $\hat{E}_m(\cdot)$ for each view V_m and therefore aim to produce tighter and more accurate surface estimates. Moving scans, via optimisation in the transform space, to find poses that result in high energy values lets us perform registration without requiring "hard" point pair correspondences, where each point is required to correspond uniquely to (typically) the closest point in another point set. Discretisation and sensor sampling quantisation may prevent exact one-to-one (true) correspondences between point sets. An ideal matching of the underlying geometries, therefore, cannot be guaranteed which may prove problematic in some problem instances for "hard" point pair correspondence based techniques. Softassign [106] and EM-ICP [110] are examples of work that addressed this problem for the case of two point sets, using weighted multi-point soft (*e.g.* probabilistic) matching and avoid forcing hard correspondences between point sets.

Forcing hard point correspondences can also prove problematic for the case of multi-view registration. Various surface representations to address this problem have been introduced such as triangulated surfaces [80], parametric representations [66] and probabilistic distributions [53, 274]. Chui and Rangarajan [57] develop an algorithm extending the early soft correspondence work of [106] to non-rigid registration. More recently [215] perform group-wise registration on multiple sets of points, using a Gaussian Mixture Model based registration. The density function and transform space search approach that we introduce for the purposes of surface approximation and view registration can similarly be considered a soft correspondence approach to multi-view registration and yet also employs the common tactic involving alternating between optimising viewpoint transforms and correspondences while keeping the other fixed.

Formally, we define an energy based on the $\hat{E}_m(\cdot)$ (Equation 11) functions that evaluate the spatial positions of all points $\mathbf{x}$ belonging to view V_m. By adapting the generic multivariate kernel density formulation (Equation 5), we build an energy function that quantifies the quality of the registration between a surface approximation S_m and points $\mathbf{x}$ belonging to view V_m.

Position $\mathbf{x}$ is evaluated by accumulating local kernel contributions $K_{m,i}(\mathbf{x})$ (Equation 10) for each sample point $\mathbf{p_i} \in \mathcal{P}$ where $\mathcal{P}$ is the set of spatial neighbouring samples of $\mathbf{x}$ in the views $\{V_n | n = 1, \ldots, M \wedge n \neq m\}$. Section 3.3.4.2 provided detail on this choice of using a finite kernel support neighbourhood. In accordance with standard kernel density estimation, our energy value at point $\mathbf{x}$ is defined as the summation of local kernel contributions. The local contribution $K_{m,i}(\mathbf{x})$ at $\mathbf{x}$ is defined using the bi-component kernel, introduced in section 3.3.4.1 and centred on neighbouring data point $\mathbf{p_i}$. In the following section we specify how these energy function evaluations are made use of to solve instances of the multi-view registration problem.

3.4 MULTI-VIEW REGISTRATION USING DENSITY ESTIMATION

Section 3.3.5 describes an energy function formulated to infer where surfaces are likely to exist using available point cloud data as evidence. In this section we detail how this energy is evaluated and minimised to perform the multi-view registration task.

Our multi-view registration approach includes three main stages as illustrated in Figure 16: (1) coarsely aligned viewpoints are provided as input, (2) non-parametric density estimation is performed on viewpoint depth measurements to determine where surfaces are likely to exist, and (3) the alignment, based on the spatial pose of each point cloud, is evaluated and optimised in relation to this inferred surface. Like [215], we decouple our group-wise registration into two iterated steps. Views are registered to the current surface approximation using rigid transforms and then the optimised view pose positions are used to update surface approximations. We make use of Quasi-Newton optimisation techniques to iteratively improve registration by optimising pose parameters in the transform space and the surface approximation is updated by re-evaluating the density estimation under the updated viewpoint positions.

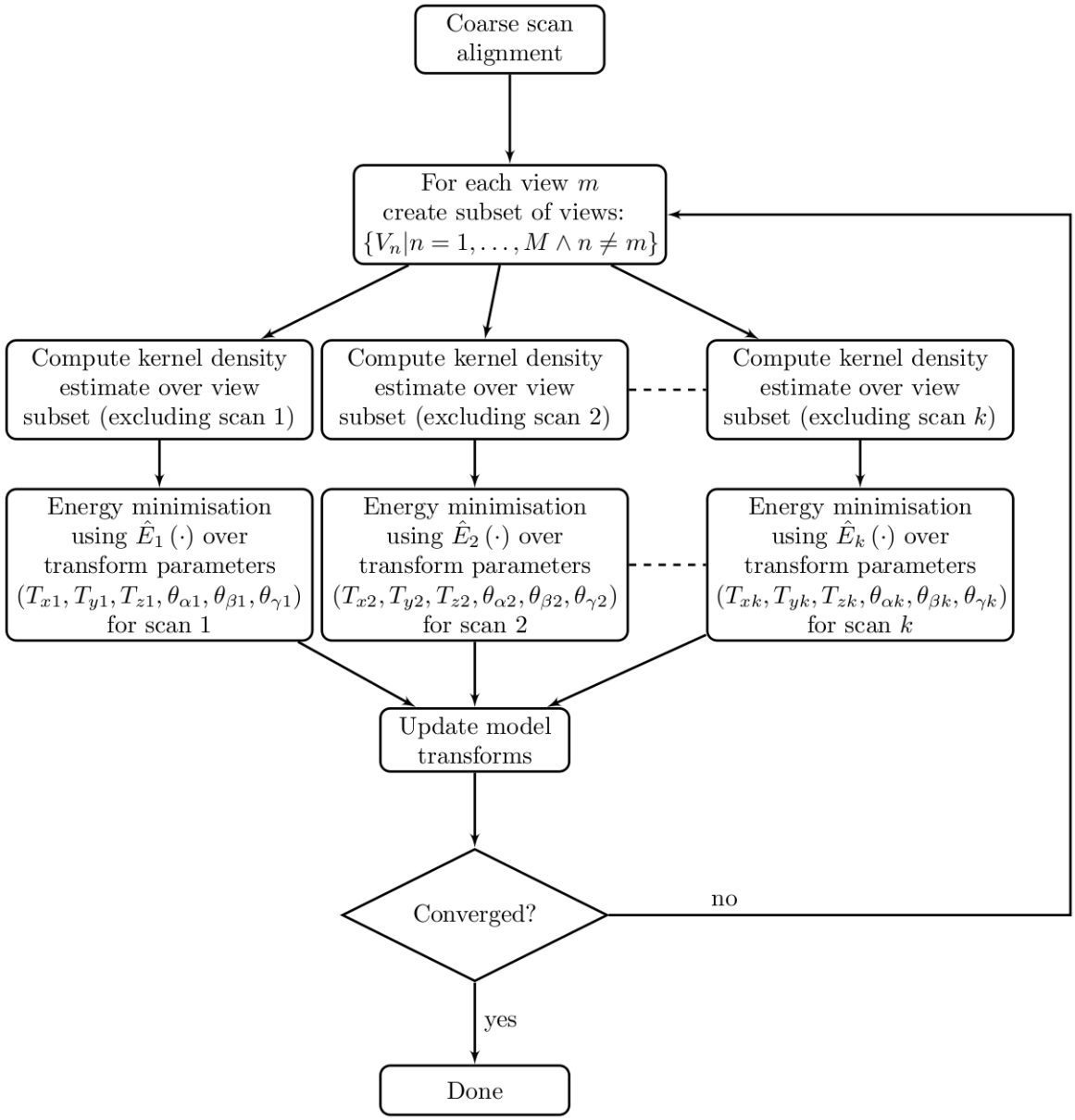


Figure 16: Our multi-view registration algorithm based on density estimation. A surface approximation is estimated for each view using density estimation over the set of remaining viewpoints. The position of each view is then independently and simultaneously optimised in the transform space. Density estimates can then be re-computed to update surfaces approximations using the updated view positions. This process is repeated to convergence.

To optimise the pose of view V_m, the density estimate defined by the set of views $\{V_n | n = 1, \ldots, M \wedge n \neq m\}$ is queried at points $\mathbf{x}$ corresponding to all spatial locations

of points in point cloud V_m. By querying the density function $\hat{E}_m(\cdot)$ at each of the member points $\mathbf{x} \in V_m$ in this fashion, the current pose of the point cloud V_m can be evaluated quantitatively. The (negative) summation of these function evaluations provides the energy to minimise when optimising the pose of V_m:

$$\sum_{\mathbf{x} \in V_m} \hat{E}_m(\mathbf{x}) \tag{13}$$

The value of each $\hat{E}_m(\mathbf{x})$ term in this summation is influenced by the current poses of the remaining viewpoints collectively. We improve the pose of V_m by searching in a 6D transform space:

$$(T_{xm}, T_{ym}, T_{zm}, \theta_{\alpha m}, \theta_{\beta m}, \theta_{\gamma m})$$

for spatial transforms that result in lower energy. This minimisation firstly evaluates the current pose of view V_m in relation to what can be thought of as the implicit surface S_m, defined by the density estimate of the other viewpoints $\{V_n | n = 1, \ldots, M \wedge n \neq m\}$. The minimisation process then searches for transforms that provide a better alignment with this surface estimate S_m. Points $\mathbf{x}$ in positions of high density (lying on or near inferred surfaces) will result in higher values and, therefore, lower energy during this optimisation process. On inspection we find that our point cloud based surface energy space is often smooth in practice (see Figures 18,19) and therefore our transform parameter optimisation search can be guided by utilising approximate derivatives $\nabla \hat{E}_m(\cdot)$ which we find via finite differencing. In practice we perform a Quasi-Newton optimisation in the rigid transform parameter space to realise this.

Using gradient information during the minimisation process, such that the energy function value is decreasing at each step, the convergence of the energy to a fixed (but possibly local) minimum is guaranteed [280]. Convergence properties for other point registration methods, such as ICP, are usually difficult to study because their cost functions, defined by *e.g.* hard nearest-neighbour correspondences, change from iteration to iteration as the point configuration evolves. In contrast, our energy function based on density estimation is defined such that each step of minimisation (within a surface estimate step – see Figure 16) decreases the same cost function.

Various registration energies have previously been optimised using numerical methods in a similar fashion [189, 154]. It is noted by [154] that for a large number of

scans, numerical optimisation may suffer from instability and slow convergence. In an attempt to avoid these problems we alternate between surface estimation and performing optimisation on the parameters of each viewpoint individually (yet simultaneously) thus keeping the parameter space optimisations low-dimensional yet implicitly accounting for the position of every other scan with the multi-view surface approximation. We apply this process simultaneously to the position of each of our M views V_m using energy functions defined by the current position of the remaining $M-1$ views $\{V_n | n = 1, \ldots, M \wedge n \neq m\}$.

Once optimal rigid transforms $(T_{xm}, T_{ym}, T_{zm}, \theta_{\alpha m}, \theta_{\beta m}, \theta_{\gamma m})$ are found for each point cloud, we apply these to each view V_m and then recompute the M surface approximations using the new collective viewpoint positions. We iterate this process of simultaneous transform parameter optimisation for each viewpoint V_m followed by surface re-estimation to convergence. In practice we can evaluate process convergence by monitoring *e.g.* (1) change in energy function values, (2) magnitude of transform parameters found at each iteration, and (3) registration error metrics (see section 3.5.1.1). Our registration algorithm is formally defined in the following pseudocode:

Input: Range scans $V_1, \ldots, V_M$
begin
  converged := 0
  while (NOT converged)
    parallel for m=1 ... M
      $S_m = \text{density\_estimate}(\bigcup_{\substack{n=1 \\ m \neq n}}^{N} V_n)$
      $\theta_i = \arg\max_{\theta} E(T_\theta(V_m), S_m))$
    end
    parallel for i=1 ... N
      $V_m = T_{\theta_m}(V_m)$
    end
    converged = test_convergence$(V_1, \ldots, V_M)$
  end
end

As scan registration improves, the local point sampling density typically becomes tighter and local kernel widths h are able to reduce adaptively to account for this. Our use of an adaptive kernel width leads to larger kernel sizes in regions of large amplitude noise due to the low sampling density and smaller kernels where scans are tightly registered. This decreases the effect of noise by reducing the contribution of noisy local maxima which in turn aids registration. We also give view cliques a high chance of intersecting during registration due to adaptive bandwidth addressing the problem of view clique point pair matching. Experimentally we observe that this results in improved registration of point sets with large scale noise and point sets that are likely to form local cliques during registration.

The registration strategy has the effect of pulling viewpoints into alignment with the inferred surface (and implicitly with other views). We iteratively update the surface estimate based on updated point cloud poses, find optimal transforms for each view and then iterate this process. Given a reasonable coarsely aligned seed, we can infer a surface (see Figure 17 for coarsely aligned scan set) without requiring view order information and we terminate the procedure either after a fixed number of steps or when energy convergence is reached (we provide convergence details in the following experimental section). In summary, this strategy provides a simultaneous global alignment strategy for multiple dense point clouds by making use of density estimation. By selecting a viewpoint merging strategy, the well registered M views can be merged into a single point cloud, providing suitable input for a surface reconstruction stage or further applications.

Figure 17: A planar slice of our energy function through coarsely aligned partial scans (Bunny data set).

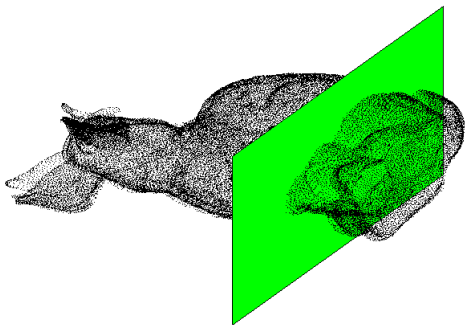


Figure 18: Energy kernel component terms. Visualisation of the planar slice through coarsely aligned Bunny data set. Left: Orthogonal projection to local plane fit kernel term. Right: Gaussian kernel term. By using the product of the components (see Figure 19 left) we are able to dampen areas of low density yet retain valid surface shape.

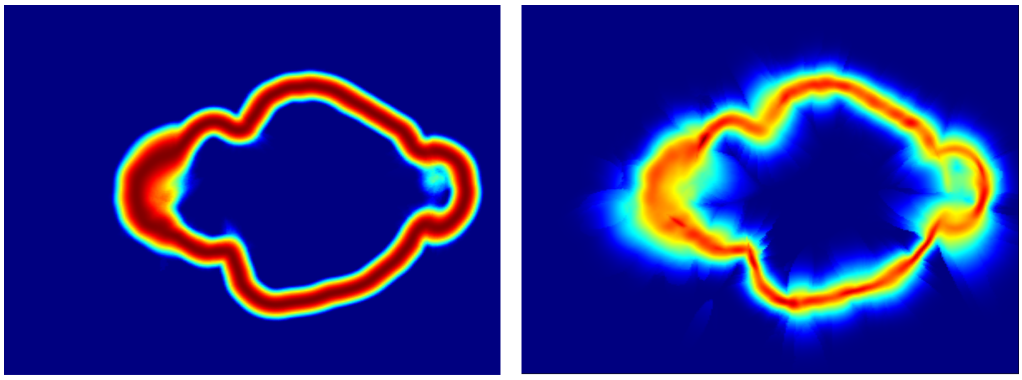
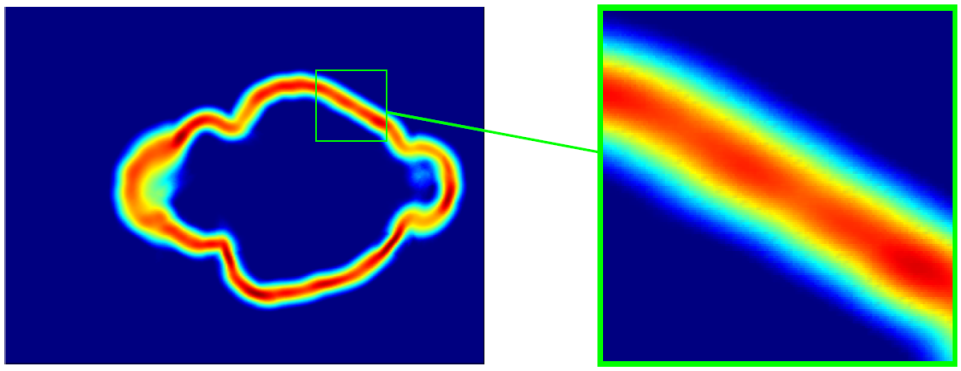


Figure 19: Our product energy function $\hat{E}(\mathbf{x})$ approximating the underlying surface defined by the coarsely aligned scans. A zoom of the slice region shows function values that are represented by colours increasing from deep blue to red. The smooth nature of our function aids the pose parameter search.



3.5 EXPERIMENTS

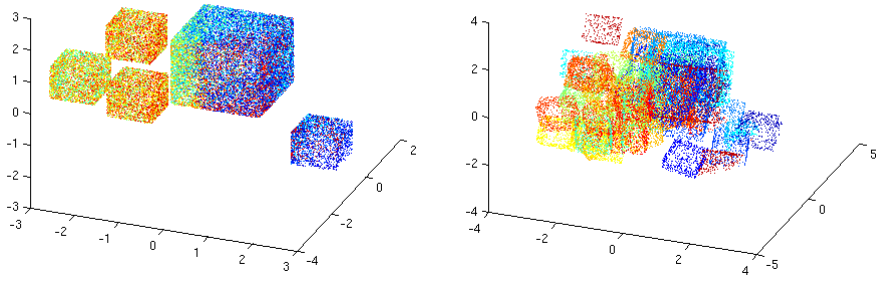
We compare view alignment results with common and recent multi-view point registration algorithms. These include a standard chain pairwise ICP [21] approach that makes use of an anchor scan and performs pairwise alignment for each pair of subsequent views. By chaining the transforms found, subsequent views can be brought into the reference frame of the anchor scan. Annealing is used to decide when convergence has been reached. Although fairly straightforward in isolation, a similar approach is often used

as an initial registration step for many applications. As an example [288] make use of this technique as an initial registration step in their cluster based surface reconstruction work. We also compare to recent multi-view registration work by Toldo et al. [262] who perform a multi-view alignment by making use of a Generalized Procrustes Analysis framework. By comparing results with other simultaneous multi-view registration work we provide analysis of how the methodology proposed here compares to state-of-the-art solutions for the multi-view registration task. For example, the techniques provided in [262] have been adopted and made use of in recent systems that succesfully address practical problems. Examples include the system proposed by [7] that is able to harness multiple consumer depth cameras to enabled 3D reconstruction of moving foreground objects. We do however concede, as noted by [263] and others, multi-view registration techniques tend to have a sparse and varied coverage in the literature. This has led to a lack of robust and fair methodology for performance assessment and comparison, making superlative claims hard to verify. When making use of real-world data sets, where ground truth alignment is not available, it becomes difficult to evaluate and quantify the results of global registration and settle for an optimal solution without resorting to intensive and time-consuming analysis of the registered views. For this reason we perform a wide range of experiments with both synthetic and real-world data.

3.5.1 *Synthetic point cloud data*

Point cloud registration experiments are carried out to systematically evaluate the proposed framework. Experimental results are compared with other recent multi-view registration work. Firstly, synthetic point cloud datasets were generated to investigate intrinsic properties of the proposed approach. Synthetic datasets provide a straightforward resource facilitating the quantitative comparison of registered view output with ground truth alignment. Synthetic data were created by generating cube and sphere-based surface models with added Gaussian surface noise. Partial views of these models were defined by simulating a camera/sensor position and sampling sets of point measurements from the synthetic surface, visible to the synthesised-camera viewpoints. If sample points are deemed visible (in the line of sight) to the simulated sensor/camera position, they are added to the point cloud of the corresponding viewpoint (see *e.g.* Figure 20 for a resulting set of point clouds).

After generating individual viewpoints by sampling from simulated camera positions, views of the synthetic data are perturbed with random $(T_x, T_y, T_z, \theta_\alpha, \theta_\beta, \theta_\gamma)$ rigid transforms (*c.* 10% of cube side length / sphere diameter translations and 10 degree rotations in magnitude) to simulate a level of coarse view alignment. The size of the perturbation aims to simulate the accuracy with which an approximate view alignment could be performed manually.



(a) Aligned ground truth pose. The dataset contains 15 partial scans with the colour of each sample point indicating the scan viewpoint that it belongs to (1 colour per viewpoint).

(b) The synthetic cube point cloud dataset with viewpoints in a perturbed pose due to random rigid spatial transforms, see text for further detail. Random perturbed poses such as this provide the input to the tested registration algorithms.

Figure 20: Synthetic cube point cloud dataset

Registration quality metrics (defined in sections 3.5.1.1 and 3.5.1.2) can be measured using the viewpoint spatial configurations obtained post-registration. Comparing algorithm view registration results with dataset ground truth alignments (readily available for synthetic data) provides an obvious assessment tool for registration quality. In the following section we briefly outline the quality metrics made use of for this task.

3.5.1.1 *Statistical error measures*

We compute standard RMS residual point pair and mean inter-point distances of the converged alignment poses. The RMS residuals are computed as the root mean square distances between the points of every view and the single closest neighbouring point from any of the other $M-1$ views. This gives a measure of the compactness of the

scans. With N points in total in the combined data set this provides N distance values $\{d_1, d_2, ..., d_N\}$ and the RMS residual is given as:

$$\epsilon_{\text{rms}} = \sqrt{\frac{1}{N}\left(d_1^2 + d_2^2 + ... + d_N^2\right)} \tag{14}$$

For the collection of M views our second RMS metric forces each sample point to identify the closest neighbouring point in *every* other viewpoint. This allocates $M-1$ distance values to each sample point in the combined data set. Many of the real-world data sets we experiment with display a non-zero (yet minor) variance in the number of point samples per view. Therefore by letting n_i define the number of points that belong to viewpoint i, we have $\sum_{i=1}^{M} n_i = N$ points in total and this second metric therefore provides $(M-1)\cdot\sum_{i=1}^{M} n_i = (M-1)\cdot N$ distance measurements $\{d_1, d_2, \ldots, d_{(M-1)\cdot N}\}$ for a set of M views such that view i contributes n_i point samples in practice. In a similar fashion as before:

$$\epsilon_{\text{group\_rms}} = \sqrt{\frac{1}{(M-1)\cdot N}\left(d_1^2 + d_2^2 + ... + d_{(M-1)\cdot N}^2\right)} \tag{15}$$

defines our second RMS metric. This secondary RMS measure is useful in addition to the first as it penalises the previously discussed "view clique" problem where scans may exhibit good local registration yet poor inter-clique registration.

The mean inter-point distance μ_{ipd} considers the average distance between each point p_i and the nearest neighbouring point p_j from all other scans combined. This once more provides n distances and we disallow pairs of points that have the same parent viewpoint. The mean inter-point distance is therefore:

$$\mu_{\text{ipd}} = \frac{1}{n}\left(d_1 + d_2 + ... + d_n\right) \tag{16}$$

This metric attempts to provide an evaluation measure of how tightly a group of viewpoints has been registered. Well registered sets of scans will typically exhibit a low mean inter-point distance.

3.5.1.2 *Estimating mean inter-point distance*

In addition to the outlined error metrics, recent work by Bhattacharyya and Chakrabarti [22] offers methods for determining the mean distance between a reference point and

its k-th nearest neighbour among points randomly distributed (with uniform density) in a D-dimensional Euclidean space. The previous section defined the mean inter-point distance between each sample point and its nearest neighbouring point from any other viewpoint as a measure of how tightly a group of viewpoints has been registered. Therefore the case $k = 1$ is considered as it is expected that our error metric utilising the mean inter-point distance μ_{ipd} (equation 16) will converge to this value when performing the registration task with datasets that contain viewpoints exhibiting (approximately) uniform object surface sampling density when in a well registered configuration.

Firstly [22] present a heuristic approach that provides a simple method for estimating the mean inter-point distance in a space containing N points. For a space containing N points we denote this heuristic approximation $\text{MeanDist}_{heur}(N)$ (see equation 17). This simple approach involves considering a unit volume of a D-dimensional Euclidean space with a density of N points. Since the unit volume contains exactly N random points (including the reference point) we divide this unit volume into N equal parts. Given that the N random points are distributed uniformly over the unit volume, each part is now expected to contain a single point. The mean distance between any point and its nearest neighbour ($k = 1$) is naively given by the linear extent of each part. Since the volume of each part is $1/N$ we expect:

$$\text{MeanDist}_{heur}(N) = \left(\frac{1}{N}\right)^{\frac{1}{D}} \tag{17}$$

It is noted that this heuristic estimate of the mean inter-point distance for the $k = 1$ nearest neighbour is a crude approximation yet provides a fast and potentially useful estimate. In [22] the authors note that values obtained by this approximation are close to the exact result only for large values of k, N and D and when the condition $N \gg k$ holds. This claim agrees with our simple 3D experimental investigation of the heuristic where we draw N points uniformly randomly in 3D space (see Figure 21, upper) and compare the measured μ inter-point distance to corresponding $\text{MeanDist}_{heur}(N)$ values (Figure 21, lower right). In the particular case pertinent to this work ($k = 1, D = 3$) we experimentally observe this approximation producing small over-estimations of the measured mean distance for the relatively small sets of point sample sizes tested ($N = 500$).

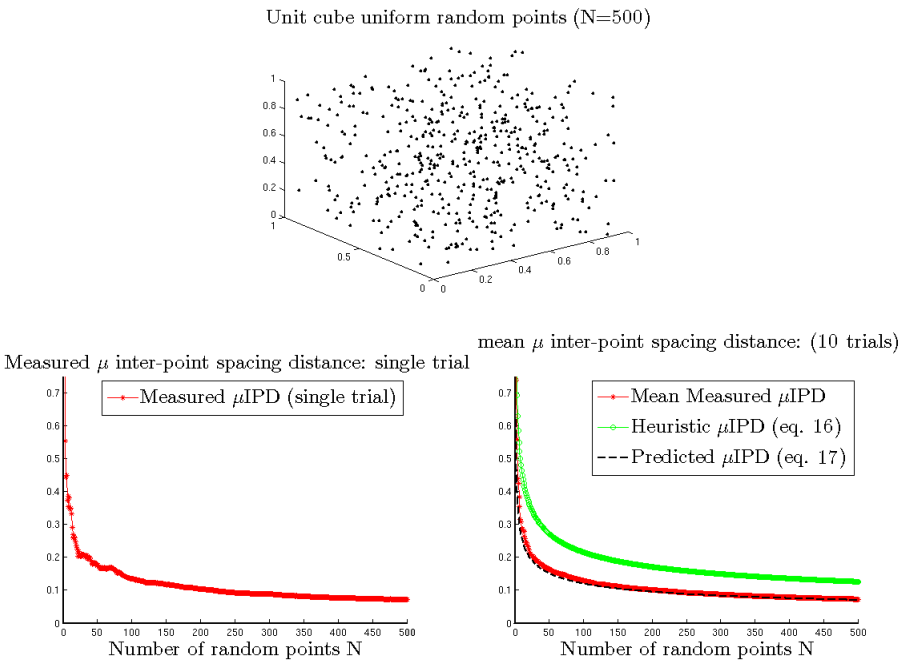


Figure 21: Top: We draw N points uniformly randomly in 3D space and measure the mean Euclidean inter-point distance. Lower left: A single trial measuring the inter-point distance drawing $N \in \{2, \ldots, 500\}$. Lower right: Average measured μ_{ipd} over 10 trials (red stars), heuristic approximation (green circles, equation 17) and exact expression (black dotted line, equation 18).

The work in [22] additionally goes on to describe a means of deriving an exact expression for the predicted mean inter-point distance. Due to the accuracy considerations outlined above and given that we are motivated by the particular case $k = 1$ and $D = 3$, this exact expressions is also investigated here. Again letting D be the dimension of a unit (hyper)cube in Euclidean space, N be the number of points randomly and uniformly distributed over the space, and defining MeanDist (D, N, k) as the mean distance to a given points k-th nearest neighbour then [22] provide an exact expression as:

$$\text{MeanDist}\,(D, N, k) = \left(\frac{[\Gamma\left(\frac{D}{2}+1\right)]^{\frac{1}{D}}}{\pi^{\frac{1}{2}}}\right) \cdot \left(\frac{\Gamma\left(k+\frac{1}{D}\right)}{\Gamma\left(k\right)}\right) \cdot \left(\frac{\Gamma\left(N\right)}{\Gamma\left(N+\frac{1}{D}\right)}\right) \tag{18}$$

For large values of N, we can make use of Stirling's approximation [109] for the Gamma function: $\Gamma\left(N+\frac{1}{D}\right)/\Gamma(N) \sim N^{\frac{1}{D}}$ therefore for large point density N in practice we can reduce equation 18 to the following asymptotic form:

$$\text{MeanDist}\left(D,N,k\right) \sim \left(\frac{\left[\Gamma\left(\frac{D}{2}+1\right)\right]^{\frac{1}{D}}}{\pi^{\frac{1}{2}}}\right) \cdot \left(\frac{\Gamma\left(k+\frac{1}{D}\right)}{\Gamma\left(k\right)}\right) \cdot \left(\frac{1}{N}^{\frac{1}{D}}\right) \tag{19}$$

where $\Gamma\left(\cdot\right)$ is the complete Gamma function (see [22] for further details). Equations 18 and 19 provide us with a reasonable estimate of the theoretical lower bound for our μ_{ipd} metric in practice (see Figure 21, lower right).

The registered point samples studied in this work tend to lie on surfaces in 3D space so aligned view sets are evaluated in relation to the estimate defined in equation 18 by asserting $D = 3$ and amalgamating all point samples of a registered view set into a single point cloud then uniformly dividing this amalgamated set into (small) spatial regions that can be considered locally planar. An octree data structure is used to achieve this spatial subdivision in practice. By assuming that well registered points will lie uniformly on small locally planar regions, it remains to count the number of points N in each (non-empty) octree region and scale the resulting $\text{MeanDist}\left(3,N,1\right)$ value by the ratio of the region (cubic volume) to the original unit cube.

By taking the mean of these $\text{MeanDist}\left(3,N,1\right)$ values over the set of small (non-empty) octree cubic regions we obtain a reasonably accurate approximation to the theoretical inter-point distance (see Figure 21 lower right) that in turn provides a sensible lower bound on registration accuracy.

This octree subdivision strategy proves a more accurate estimate than both (1) the previously introduced heuristic inter-point distance estimate (equation 17) and (2) measuring the $\text{MeanDist}\left(3,N,1\right)$ over a single unit cube bounding box encompassing the entirety of the registered views (registered view point samples are typically far from uniformly distributed in such a bounding box space). In summary equations 18 and 19 offer a useful indication of how well the view registration task has been performed in practice. By comparing experimental registration results to this limit it can be ascertained how close to a theoretically optimal view alignment has been achieved.

3.5.1.3 *Synthetic data: Registration quality experiments*

For synthetic datasets containing cube like structure, we perform registration experiments by perturbing the view set (containing 15 views) with random rigid transformations and then applying both the proposed view registration algorithm and the Procrustes method [262]. By measuring the mean inter-point distance of the view set iteratively after each rigid transform step, the registration progress is assessed. Experimentally we perform 20 trials involving randomly perturbed view set starting configurations (Figure 20b exhibits a typical starting configuration created by perturbing the ground truth pose found in Figure 20a). We find 20 trials sufficient to obtain statistically significant results and provide further detail in section 3.5.1.4. In Figure 22 we plot mean and standard deviation μ_{ipd} progress for the measured inter-point distance averaged over 20 trials for both algorithms. Since the ground truth alignment is available we are able to measure the μ inter-point distance of the ground truth pose and compare this to the converged method values and with the theoretical lower bound provided by equation 18.

For synthetic datasets, where ground truth alignment is available, it can be observed that the theoretical lower bound (equation 18) underestimates the measured μ_{ipd} value of the ground truth pose by $\sim 10\%$ (see Figure 22). We propose that this discrepancy may be due to the granularity of the spatial octree subdivision strategy chosen to evaluate the aligned view set. Given the relatively small discrepancy, comparing converged μ_{ipd} results to this lower bound approximation can be considered a valid assessment of registration quality for datasets where no ground truth alignment is available. For the simple synthetic datasets experimented with in Figure 22, the proposed multi-view registration strategy consistently converges to μ_{ipd} values closest to the measured ground-truth pose μ_{ipd} (and also closest to the introduced theoretical lower bound) experimentally. The coarse seed alignment for each trial is created by perturbing the ground truth alignment with random rigid transformations and due to the number of intermediate transforms found and applied by the compared methods varying, the horizontal step axis is rescaled so timing comparisons are not valid but convergence behaviour is. Visual assessment of the resulting view poses in comparison to ground truth pose is carried out in the following section (section 3.5.2.2).

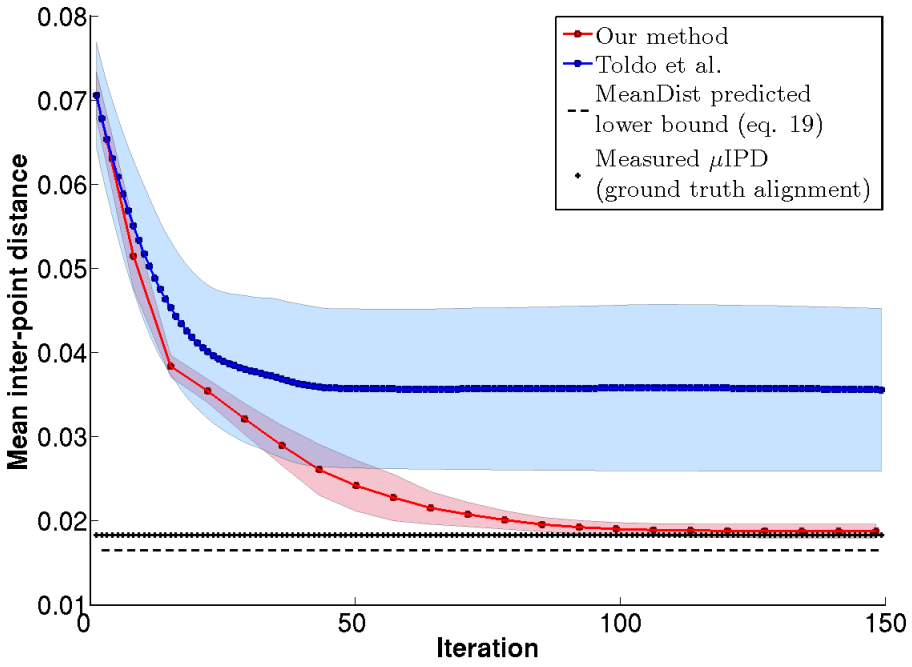
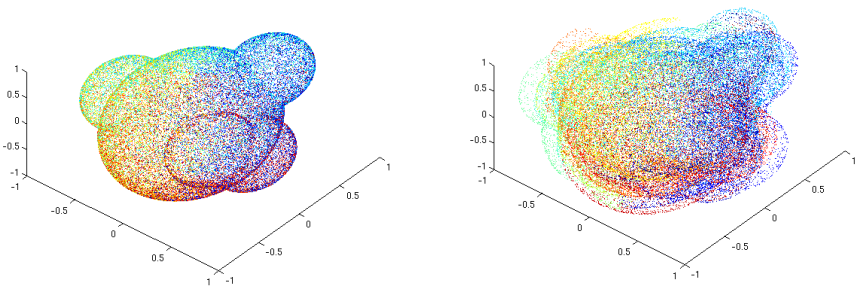


Figure 22: Mean inter-point distance during registration of synthetic cube data set. Horizontal step axis rescaled so timing comparisons are not valid but convergence behaviour is (see text for details). Measured mean inter-point distance and ±1 standard deviation plotted for 20 repeated trials between compared methods. The consistent μ_{ipd} value measured for the ground truth view pose (black '+') and predicted inter-point distance (equation 18, black dotted line) are plotted for comparison.

3.5.1.4 *Synthetic data: Registration robustness experiments*

Experiments are performed with a synthetic sphere-like, bulbous in shape data set to investigate the robustness of our method. A repeat experiment was carried out by seeding the synthetic data with random sets of pose perturbations and assessing alignment algorithm performance on these sets of random seed positions. Seed positions were again obtained by perturbing each scan from a set with random $(T_x, T_y, T_z, \theta_\alpha, \theta_\beta, \theta_\gamma)$ transform parameters such that the seed positions resembled coarse manual alignment. An example perturbed seed position for the synthetic sphere data can be found in Figure 23b with the ground truth alignment found in Figure 23a.

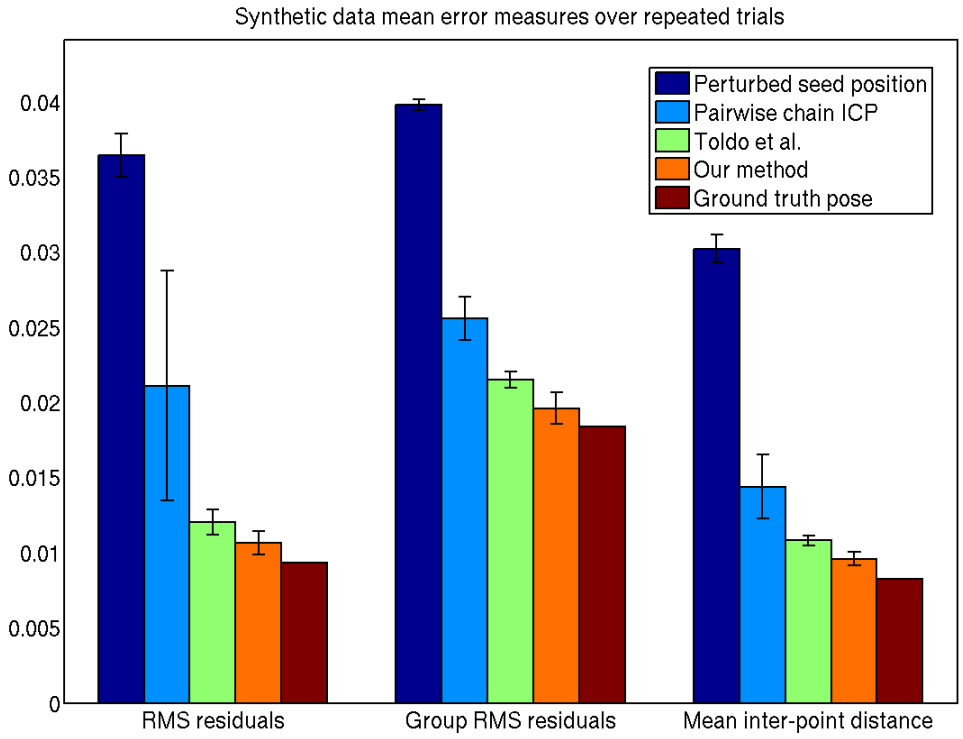


(a) Synthetic sphere-like bulbous shape partial depth scans. Twenty viewpoints in ground truth alignment with one colour per viewpoint.

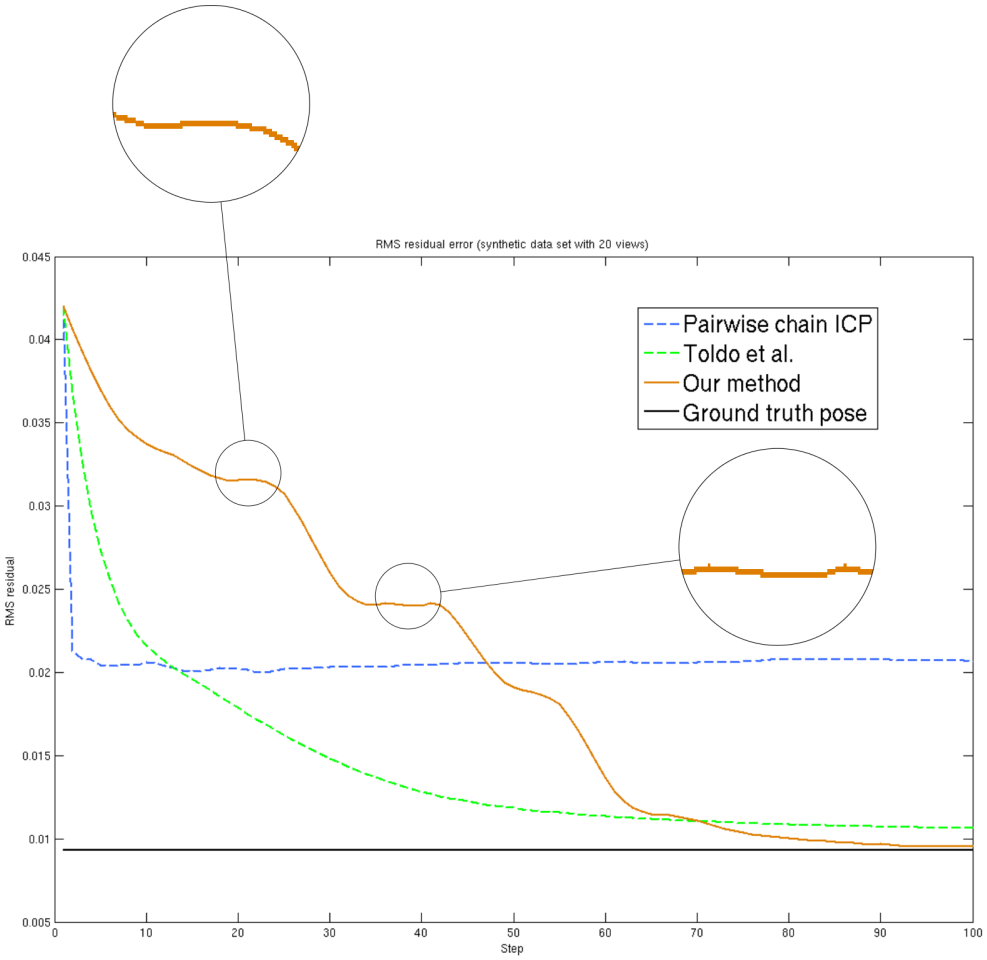
(b) Each viewpoint perturbed by a random rigid transform (typical input seed alignment for the registration algorithms). The level of perturbation attempts to simulate coarse manual scan alignment.

Figure 23: Synthetic sphere-like point cloud datasets.

This experimental work attempts to provide insight into basins of convergence. It involves exploring which algorithms are able to converge consistently and how often gross alignment errors or failure to converge to a reasonable solution are likely to occur. The synthetic sphere-like data is initialised with 20 different seed positions and the alignment results, produced by the three considered registration algorithms, are compared using the error measures introduced previously (section 3.5.1.1). We report the three measures averaged over 20 seed positions for each of the three alignment methods and also report mean seed position and known ground truth pose metrics. Error bars indicate one standard deviation of the repeated trials. Results are found in Figure 24a. We note that values resulting from the Procrustes alignment method [262] are again similar but inferior to our method. We perform a simple paired two-tailed t-test on the post-registration metrics from the Procrustes alignment samples and those from our method. We find that all three of our error metrics obtain statistical significance at the $p \leq 0.001$ level between these techniques. We note that the mean differences (effect size) between the methods on these metrics is small $(0.0013, 0.0019$ and $0.0012)$ and our $N = 20$ is relatively low. However we observe that the proposed framework consistently produces lower values in almost every trial, leading to the low p values. A larger number of trials with more complex synthetic data sets would give more power to the conclusion and provide further robustness evidence.



(a) Mean values for our three error measures across 20 registration trials on our sphere-like synthetic data set. Mean seed position and ground truth positions are also measured for comparison.



(b) RMS residuals evaluating algorithms using the sphere-like synthetic data set. We display measured RMS residual values at each transform step for the compared methods and the consistent RMS residual value measured using the ground truth pose (solid black line) for comparison. See text for additional discussion, including plateauing convergence behaviour.

We further analysed the synthetic sphere-like data results by computing RMS residuals at each intermediate transform step of the registration progress (Figure 24b). For each method, we measure RMS residuals after each spatial transform is applied until convergence is reached. Give that the number of intermediate transforms applied by each method varies, for display purposes, we rescale the horizontal step axis to $0-100$ so timing comparisons are not valid but convergence behaviour is. We do not compare wall-clock run times directly as the algorithm we propose takes advantage of a multi-core parallel implementation (see Chapter 4 for further multi-core implementation details) such that the work of individual viewpoint alignment is distributed simultaneously to multiple processors in practice. Contrastingly, the additional registration techniques we compare to here are implemented in a serial fashion. The synthetic data set experiment found in Figure 24b converges on the order of minutes in each of the three cases examined, but we note that the proposed algorithm is making use of more computational resources due to the transform space search technique employed and the distributed implementation.

The global residual of the Procrustes algorithm [262] is comparable to our approach but converges to a weaker solution on our synthetic sphere dataset in 17 of the 20 trials performed. Both the Procrustes algorithm due to Toldo *et al.* and the chain ICP methods exhibit fast initial RMS error convergence (using this dataset) by pulling the viewpoints close together yet, particularly in the case of the ICP method, they plateau at suboptimal solutions. This is in agreement with the previous μ_{ipd} solution quality convergence experiments performed in section 3.5.1.3. We additionally note that the convergence of our method exhibits periodic plateauing behaviour for this data set. This can be explained by the fact that we periodically re-estimate our density based surface before iteratively optimising viewpoint locations. Viewpoint optimisation may converge for a given surface estimate, but once the surface estimate is updated using optimised scan positions the surface estimate typically becomes tighter and more confident such that further optimisation is possible. This process typically aids registration and improves alignment accuracy.

For tasks where final registration accuracy is of prime importance one could initialise alignment by performing a single pairwise ICP iteration before switching to our technique. Exploring this possibility provides one avenue for future work. The proposed approach converges to the best final solution in terms of closest to the ground truth

RMS value in the majority (17/20) of the experimental runs on the synthetic data sets generated.

Our synthetic data sets are straightforward in construction and provide only simple surface structure. They are however a useful tool in terms of assessing how close to a ground truth position a registration algorithm is able to achieve across multiple trials. Averaged across sets of 20 runs, the proposed framework consistently comes closest to the introduced theoretical metrics (section 3.5.1.2) providing initial evidence in support of the claim that, of the tested methods, the introduced method is able to achieve a view pose configuration closest to a theoretically optimal registration. The introduced method also displays a wide basin of registration convergence as supported by the evidence that the method reports values closest to the ground truth pose alignments across the statistical error measures (section 3.5.1.1) investigated. Performing additional experiments of this nature that involve increasing the complexity of the synthetic data surface structure would provide more evidence to support these claims and therefore provides one potential area for further work.

3.5.2 *OSU laser database*

Additional experimentation is performed by making use of real data sets consisting of laser range scans from the OSU/WSU Minolta laser database [192]. The OSU viewpoints are produced by a laser scanning process and the subjects made use of here include: “Angel”, “Bird” and “Teletubby” figurines and a spray bottle (“Bottle”). Figures 29-32 show these datasets. Each scan viewpoint is composed of between 2500 and 7000 points (pre-processing involved sub-sampling, for computational reasons, the views to 50% of their original sample points). OSU data sets are obtained by real-world acquisition, noise is intrinsically present and object point sampling is not necessarily uniform across views.

The object test sets contain between 11 and 20 viewpoints each and a summary of the data set properties and kernel bandwidths we use for registration experiments are given in Table 2. The k-neighbourhood (influencing kernel bandwidth) is chosen for each data set such that k is 0.5% of the total number of points in the data set. This method of choosing k results in a varying k value for each data set but the variance of *inter*-scan point sample magnitude within each set of explored (intra-)object views is

Table 2: OSU and synthetic data set statistics

Data set	Number of viewpoints	Mean points per view	Bandwidth size k-neighbourhood
Angel	18	6314	$k = 560$
Bird	18	4521	$k = 400$
Bottle	11	2883	$k = 160$
Teletubby	20	2671	$k = 270$
Synthetic spheres	20	4672	$k = 460$

low. Therefore by choosing k in this fashion, we find that our resulting bandwidth h is typically on the order of one to ten times the mean inter-point Euclidean distance of the coarsely aligned input data (dependent on local alignment and sampling density). Empirically, this proved to be a reasonable rule for selecting k. An obvious extension would involve investigating more principled methods for selecting k (*e.g.* [193] recently investigated selecting optimal KNN k values for *univariate* kernel density estimation bandwidth selection). A different strategy would also be required for data sets containing large point sample magnitude variance between viewpoints. Experimentally we found that 150 - 600 neighbours per local kernel was suitable in practice for the data sets explored. We include the exact point neighbourhoods used for our experiments in Table 2.

3.5.2.1 *OSU database bandwidth selection experiments*

Varying the percentage of total points used to define the kernel k-neighbourhood size is explored (see Figure 25). Making use of the "Bird" OSU dataset, we provide evidence that final object registration quality is not overly sensitive to the value of k selected, providing some support for the robustness of this simple bandwidth selection strategy.

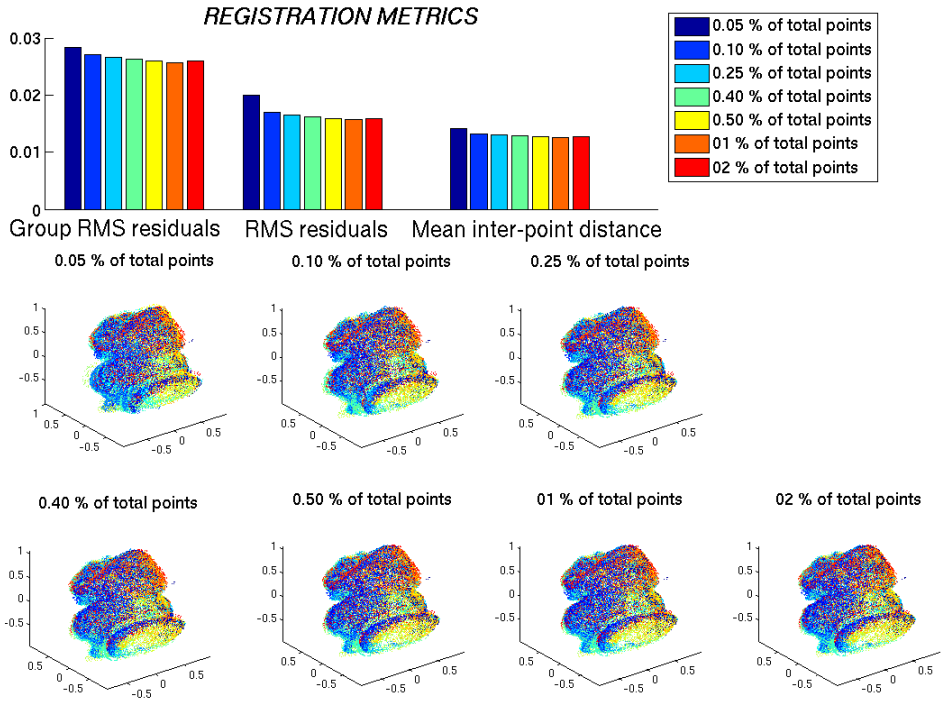


Figure 25: Sensitivity investigation of the proposed kernel bandwidth selection strategy. Stable error metric values provide some evidence that final object registration quality is not overly sensitive to values of k selected (see text for further detail).

We perform experiments with OSU data sets in order to provide justification for our kernel bandwidth selection strategy. As noted previously, the chosen method for selecting kernel bandwidths involves adaptively defining a k-neighbourhood as a percentage of the total points of the data set to align. For the experiments previously documented in this chapter we consistently make use of 0.5% percent.

In Figure 25 the percentage of total points used to dictate the kernel bandwidth k is varied when performing the alignment task with the OSU "Bird" data set. Viewpoints are seeded in identical, coarsely aligned states (see Figure 30, left-most column, for examples of hand aligned seed configurations). Identical coarse seed alignments are provided to our algorithm such that repeated multi-view registration can be performed whilst varying kernel bandwidth parameters.

We use the kernel surface approximations defined by the differing kernel sizes to perform comparable registration tasks. Starting from identical initial view configurations (a typical coarse configuration is provided by manual alignment) and fixing the number

of transform parameter optimisation rounds in each experiment to 10, we evaluate the final configuration of the set of "Bird" view in each case using the three error metrics outlined previously.

The variance of the three quality metrics studied can be seen to be low across the range of k-neighbourhood sizes explored (Figure 25). Stable error metrics values provide some evidence that final object registration quality is not overly sensitive to the value of k selected. This robustness in turn provides some evidence supporting the choice of this simple k selection strategy. Values of k explored in this experiment were defined by using between 0.01% and 2% of the total points in the data set. This corresponded to k values in the range 5 – 900 for the "Bird" data set. As noted earlier, a promising line of further work would involve investigating more principled methods for selecting k (such as those explored by [193] for univariate densities).

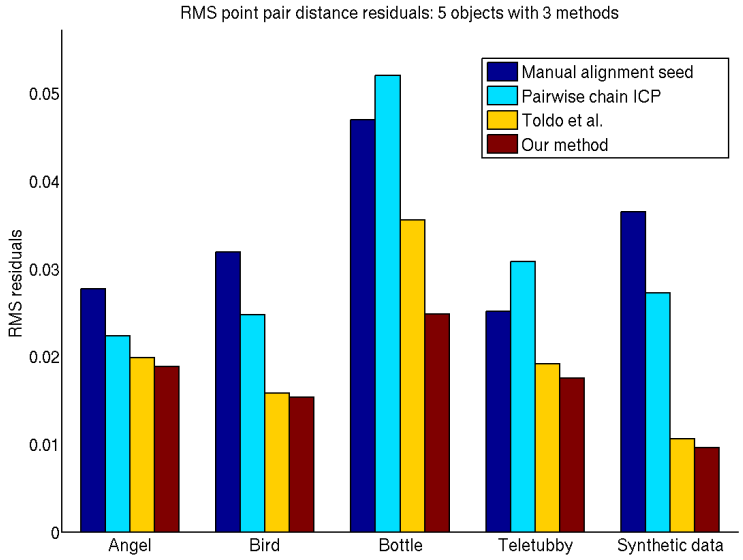
3.5.2.2 *OSU database registration experiments*

Analogous to synthetic dataset simulated coarse alignments, prior to registration, the OSU datasets were coarsely hand aligned but some misregistration is still evident (see Figures 29 to 32). We analyse the results by examining the three statistical error measures introduced previously (section 3.5.1.1).

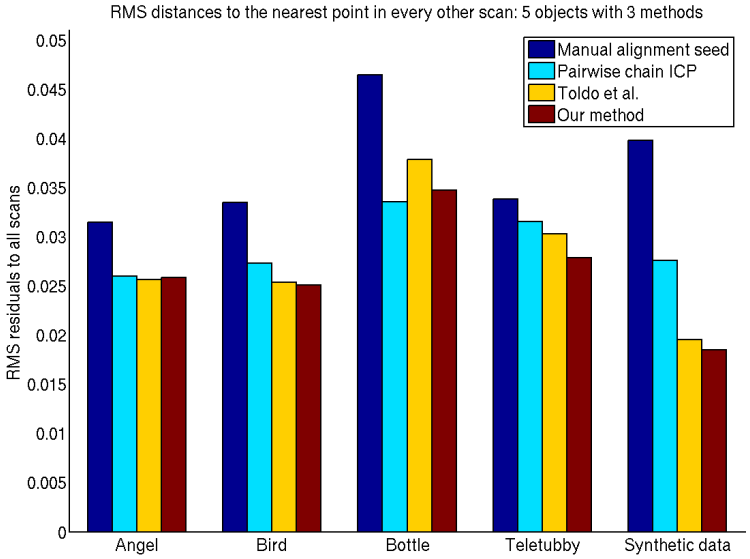
Information detailing the sequential order of view capture is not required by our approach or that of the Procrustes method of Toldo *et al.* [262] however the pairwise ICP technique does require this information because pairwise ICP alignment depends on input viewpoints exhibiting non-zero overlap (a minimum of $\sim$ 30% view overlap was found to be a necessity for registration success experimentally).

We apply the introduced statistical error measures to the resulting alignments generated by the three registration methods evaluated. Figure 26 shows the experimental results. The approach introduced in this work performs best in terms of our RMS residual and inter-point distance evaluation metrics in the data sets experimented with. In two cases the pairwise chain ICP technique converges to a solution that increases its residual error metric above the baseline hand alignment. These cases exhibit some reasonable pairwise scan alignment but global object shape is poor. In contrast, the group RMS value corresponding to applying the ICP method to the "Bottle" data set is found to be the lowest of the three techniques. This is due in part to what we call "over merging" of scans. Views are drawn together as a group but the original object

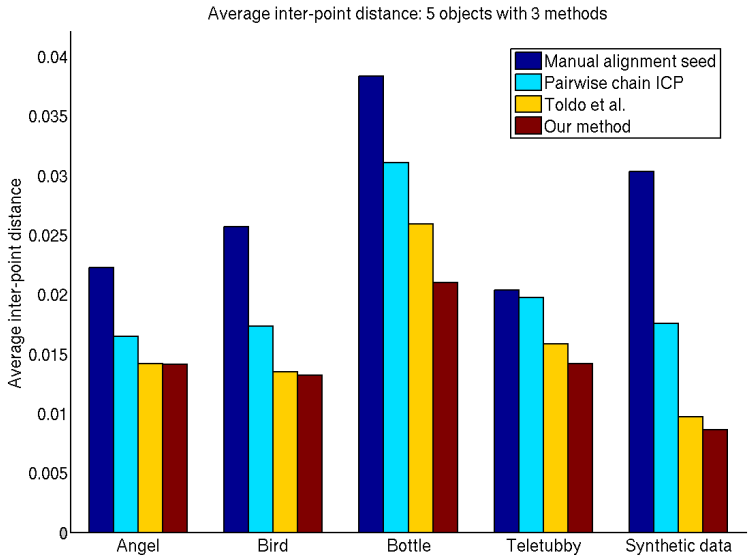
shape is detrimentally affected as scans are only being registered in a pairwise fashion. Visually inspecting the alignment in this case provides evidence that the ICP result is not optimal. With this data set our method provides an improvement in the remaining two statistical measures and a visually improved registration (see Figure 27).



(a) RMS residuals on converged OSU and synthetic data sets



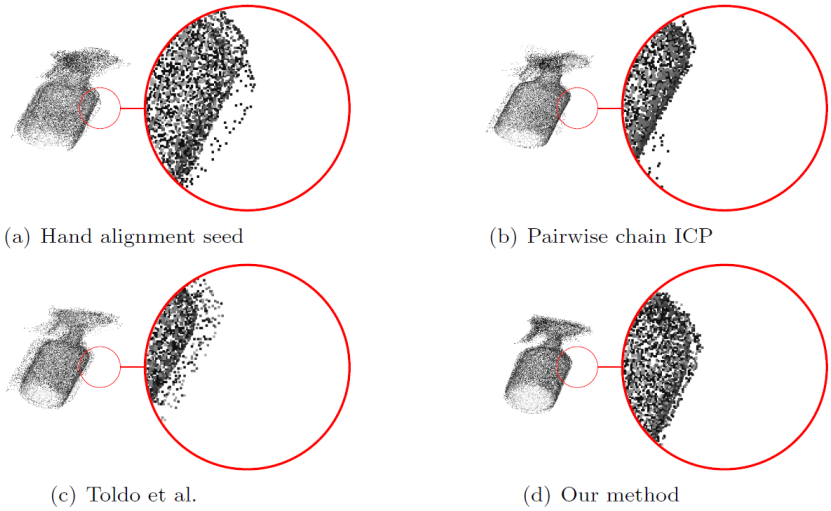
(b) Group RMS residuals on converged OSU and synthetic data sets



(c) Mean inter-point distances on converged OSU and synthetic data sets

Figure 26: Registration metrics

Initial scan configurations and final alignments pertaining to the investigated methods are shown in Figures 29 to 33. Our technique is able to converge to an acceptable minima for each data set investigated, however the pairwise ICP method in particular exhibits relatively large failure modes in some cases. In particular, the ICP technique does not find acceptable alignments for the "Bird", "Bottle" and synthetic data experiments. The Generalized Procrustes Analysis technique in general fares well yet also exhibits some failure with the "Bottle" data set explored here. We confirm the findings of [262] that, applied to multi-view registration problems, sequential ICP based algorithms require the additional information that view order is known *a-priori* yet exhibit results that are generally worse than more recent simultaneous optimisation techniques.



(a) Hand alignment seed (b) Pairwise chain ICP

(c) Toldo et al. (d) Our method

Figure 27: The OSU "Bottle" data set converges to similar acceptable poses using all three registration techniques however the chain alignment technique has partially collapsed the desired object shape as a result of attempting to minimise pair-wise distances. The proposed technique makes use of KDE in an attempt to infer global object shape information. In the example shown an improvement to the global alignment of the object is evident.

In two of the OSU data sets ("Angel" and "Bird") the final RMS residuals and mean inter-point distance values our method produces are very similar to the values resulting from the Procrustes alignment method [262]. In these cases the geometrical registration results are also visually similar and both methods exhibit good fine registration for these data sets although some differences are evident on applying a post-registration surface reconstruction (see following section 3.5.3).

3.5.3 *Surface Reconstruction*

The goal of surface reconstruction, as defined by [145], is to determine a surface S' that approximates an unknown surface S, using a sample $\mathcal{P}$ and possibly information about the sampling process. Achieving this goal in an important, well studied fundamental problem in geometry processing and often uses point cloud data as the input sample. Most reconstruction methods can be classified as either an explicit/parametric (*e.g.* triangulation based) or implicit (*e.g.* level surface $f(x, y, z) = c$) based surface representation. Implicit methods are an important class of reconstruction technique as they tend to offer topological flexibility and robustness to sensor noise. However, these methods often require points supplemented with normal information to be able to reconstruct surfaces. When reconstructing implicit surfaces from data acquired from multiple views, e.g. from laser scans, accurate fine registration is especially important if point normals are not provided by the scanning technology. Alternative normal acquisition typically involves estimation using adjacent nearby points. It is therefore not practical to apply such surfacing methods to multi-view data sets that contain significant view registration errors. The registration process applied prior to constructing implicit surfaces from sets of multi-view data is an area of current research and provides a further assessment for our registration framework.

We apply our registration technique to sets of multi-view point clouds and then reconstruct a surface from the aligned data. For comparison we also reconstruct surfaces from the coarsely aligned input point clouds and the final viewpoint positions provided by the alternative methods that were introduced at the start of section 3.5. For surface reconstruction we use a well known implicit reconstruction technique, Poisson surfacing [145]. Poisson surfacing computes a 3D indicator function χ (defined as 1 at points inside the model and 0 at points outside, as dictated by the point surface normals), and then obtains the reconstructed surface by extracting an appropriate isosurface. Since Poisson surfacing requires oriented normal information at each point, this provides a suitable method to test the alignment quality of our method. We estimate point surface normals by fitting a plane to the k-nearest-neighbour points (for $k = 10$) in the aligned view sets and propagate coherent normal directions from an arbitrary starting point using a user defined camera viewpoint to influence the indicator function χ. Surfacing result comparisons are shown in Figures 34-38.

Applying a surfacing method directly to the coarsely hand aligned data often produces gross reconstruction failures as might be expected. The surfacing technique alone is often not able to recover appropriately from the relatively poor registration provided by our hand aligned data sets. This is especially evident in our "Angel", "Bird" and synthetic data experiments where poor alignment causes gross errors and unsmooth surfaces. Visual flaws are also evident in the surfaces that result from point clouds aligned using the simple chain ICP method in the cases of the OSU data sets. In particular results from the "Angel" data set exhibit the failure of the simple ICP method to faithfully reconstruct the wing portion of the model. We argue that this can be attributed to the minor yet evident misregistration during the alignment process. The Procrustes algorithm [262] generally provides good input for surface reconstruction and the "Angel" and "Bird" data sets provide surface results that are visually very similar to ours. Our method produces slightly better quality limb reconstruction of the "Angel" data set however some small geometrical errors are still present in both results. The resulting model from the "Bird" data set using the Procrustes algorithm and our method are also very similar yet our method provides small visual improvements to areas of intended high smoothness such as the feet. Enlarged version of the surfaced point sets for these results can be found in Figure 39. The "Bottle" and "Teletubby" data sets exhibit significant surface reconstruction failure from the input provided by the Procrustes method yet fair better when using our technique. In conclusion, the results of applying a surfacing method to the registration results provided by our method tend to show visually improved reconstructions in the data sets experimented with.

3.6 EXPERIMENTAL SUMMARY AND DISCUSSION

Registration experiments are performed across multiple data sets evaluating results visually and with statistical error measures. A varying range of points per data set has little effect on the capability of the proposed method, working well across the range of point cloud sizes. An experimental set up, making use of both synthetic and real data sets, demonstrates the robustness and accuracy of the proposed method in relation to common and contemporary work for the task of simultaneous multi-view registration. The proposed registration framework is able to demonstrate quantitative results that are, in many cases, better than start-of-the-art approaches for this task.

Multi-view scan registration is typically cast as an optimisation problem. The error landscape depends on the type of data being registered, outliers, noise and missing data. As noted by [252] and observed in our experiments, if the surfaces are relatively clean and there is a good initial estimate of alignment then local optimisation such as using an ICP based method is an efficient choice. However, if there is significant noise, the initialisation is poor or the view order unknown then these methods may not converge.

When the view ordering $V_1, ..., V_M$ is known, registration can be performed pairwise between consecutive views and global registration can be obtained by concatenating the obtained pairwise transformations. As we observe experimentally, even when all pairs are apparently well registered, lack of global optimisation can result in misalignments at the stage of full model reconstruction due to registration error accumulation and propagation.

In this work we propose a novel technique to tackle the task of simultaneous alignment of multiple views. By attempting to solve simultaneously for the global registration by exploiting the interdependence between all views we implicitly introduce additional constraints that reduce the global error. We base our approach on well established kernel density estimation theory.

We have shown that our technique is capable of aligning depth scan sets with real-world noise amplitudes from seed alignments that are only coarsely defined. We demonstrate the capability of our algorithm on synthetic and real-world data sets captured using laser scanners. Further to this we show that our approach can be used in conjunction with a surface reconstruction method [145] and produce surfaces for visualisation purposes. Figure 28 provides an example of how our algorithm fits into an object reconstruction pipeline when starting from unaligned depth measurement data.

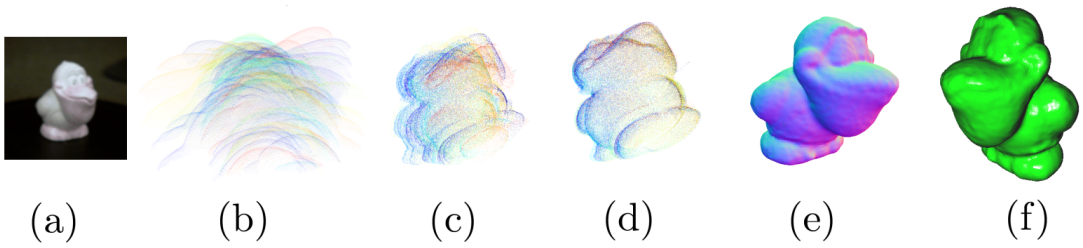


Figure 28: (a) RGB data from Ohio State University ("Bird" set) (b) Partial depth scans from OSU (c) Pre-energy minimisation (coarse alignment by hand) (d) Multi-view registration performed with our method (e) Meshed with normal orientations (f) Phong shaded Poisson surface

Methods such as the one proposed here, making use of non-parametric density estimation of spatial measurements, are able to handle the reconstruction of objects exhibiting arbitrary geometrical complexity but contain no special handling of sharp features such as might be commonly exhibited during measurement of *e.g.* mechanical or machined parts. Related work addressing sharp features has been introduced by [114]. Incorporating such considerations into our registration framework provides an avenue of interesting future work. A related potentially promising areas of further exploration include diversification and variation of point cloud dataset size and structural complexity. Additionally our method allows every scan view to converge independently to a maxima of the proposed energy function, so parallelism at the depth scan level provides a potential near linear speed-up of the registration process enabling application of our registration framework to extremely large data sets. In the following chapters these ideas are explored further and evaluation is carried out on sets of many range scans *e.g.* 100's (see Chapters 4 and 5).

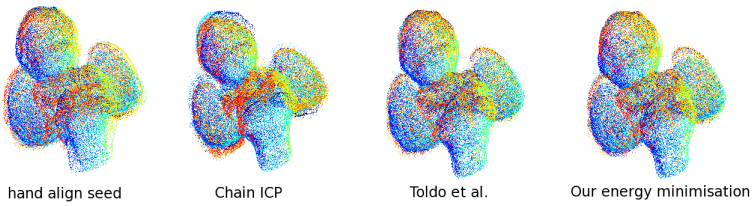


Figure 29: Angel data set final position comparison.

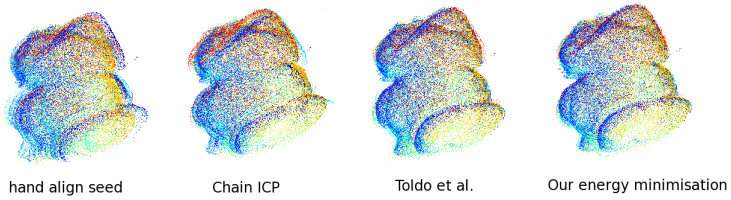


Figure 30: Bird data set final position comparison.

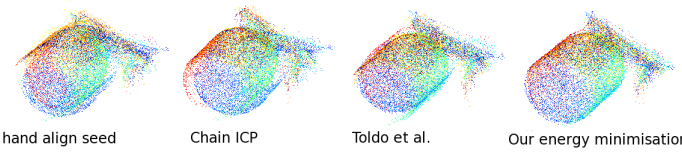


Figure 31: Bottle data set final position comparison.

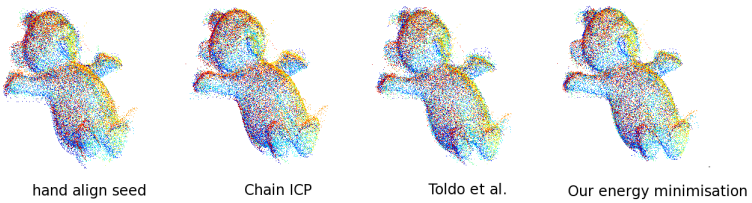


Figure 32: Teletubby data set final position comparison.

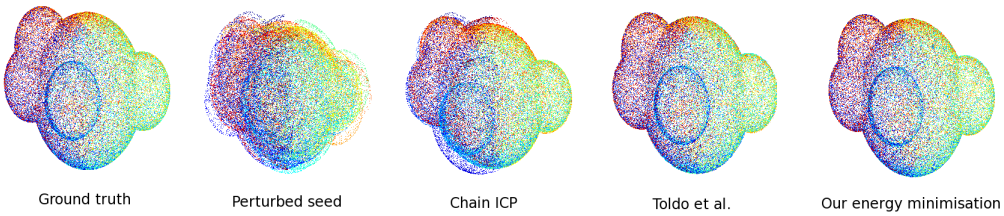


Figure 33: Synthetic data set ground truth and final position comparison.

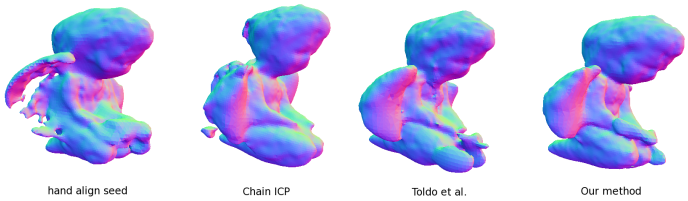


Figure 34: Angel data set. Poisson surfacing applied to final configurations.

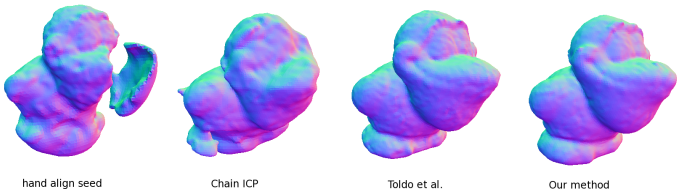


Figure 35: Bird data set. Poisson surfacing applied to final configurations.

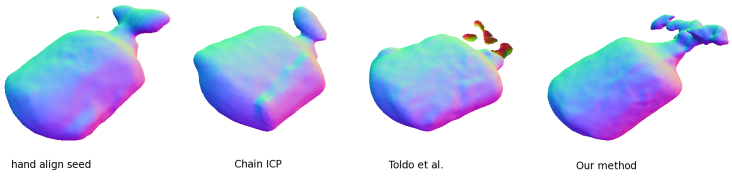


Figure 36: Spray Bottle data set. Poisson surfacing applied to final configurations.

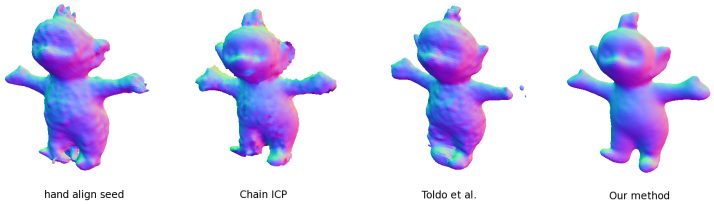


Figure 37: Teletubby toy data set. Poisson surfacing applied to final configurations.

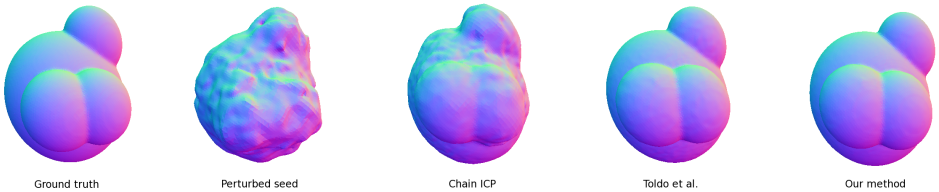
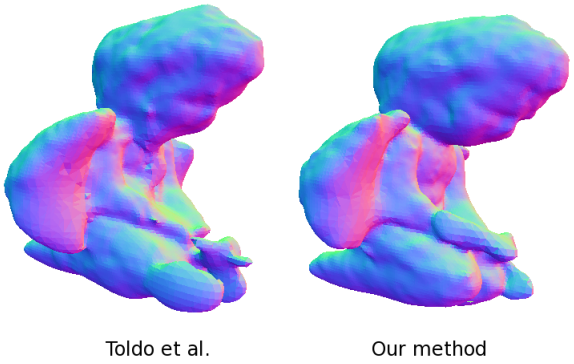
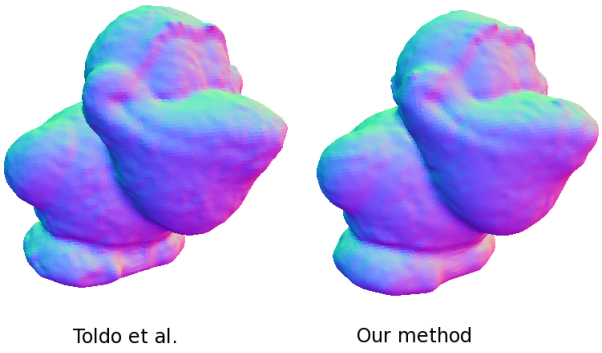


Figure 38: Synthetic data set. Poisson surfacing applied to final configurations.



(a) “Angel” registered point clouds with Poisson surfacing applied to final configurations.



(b) “Bird” registered point clouds with Poisson surfacing applied to final configurations.

Figure 39: Enlarged versions of “Angel” and “Bird” data sets with Poisson surfacing applied to final configurations.

Part IV

SEMI-SYNCHRONISED TASK FARMING

4

SEMI-SYNCHRONISED TASK FARMING

4.1 INTRODUCTION

In the previous Chapter a novel approach to multi-view point cloud registration was introduced and properties of the method were experimentally evaluated using small, multi-view, point cloud datasets. One of the key features pertaining to the proposed technique was that viewpoints are aligned *simultaneously*. This feature allows depth sensor scans, in the form of 3D point cloud data, to be considered and registered in parallel. Simultaneous registration techniques can often be considered demanding in terms of computational expense when compared with traditional serial alignment approaches. By considering all views simultaneously typically an increased computational cost is incurred as these approaches must, at each iteration, compute some registration error between each range view and some form of reference. A solution to the multi-view registration problem, capable of handling large data sets, consisting of many viewpoints, therefore provides a good candidate for parallelised implementation.

The registration techniques introduced in this work update view poses using non-linear optimisation in the pose transform space. Like many simultaneous registration strategies, this approach is expensive if attempting to align many viewpoints. For large instances of the problem that this approach aims to solve, additional computational expense may be tolerated when high quality results are considered a priority (a property considered common to many tasks in the field of computer vision and beyond). If however maximising performance in terms of *e.g.* minimising run time or response time is a prime concern, such as with systems expected to operate in real-time, then one obvious route of further enquiry involves investigating the ability to harness distributed

or parallel computation able to take advantage of the simultaneous aspects and nature of the introduced registration framework.

Towards these time performance based goals, this Chapter firstly introduces a generic task farming framework that we call *Semi-Synchronised Task Farming* (SSTF) and goes on to provide detail of how our multi-view registration procedure can be implemented under this strategy. Our multi-view registration task serves as an example to illustrate how applications that exhibit potentially distributable components can be implemented under this task farming strategy. Semi-synchronised task farming splits a given problem into a number of stages. Each stage involves firstly distributing independent tasks to be completed in parallel. This task set, comprised of many individual tasks, may require some form of inter-task communication upon all tasks completing. This is realised in practice by a set of synchronised global decisions, based on information retrieved from the distributed results, being made upon task set completion. The results influence the following task distribution stage. This task distribution followed by a result collation process is iterated until overall problem solutions are obtained.

Performance models inspired by the BSP (Bulk Synchronous Parallel) model [267] are also introduced that allow for accurate run time prediction of distributed algorithm implementations and this in turn enables predictions of expected gains over serial implementations. The quality of these predictions is assessed by extensive experimental analysis of the distributed algorithms implemented under the task farming framework. We construct a model to formalise our task distribution framework and with this formalisation, our model provides overall task completion time predictions. Experimental benchmark results comparing the performance observed by applying our framework to solve real-world problems on compute clusters to that of solving the tasks in a serial fashion are presented. By assessing the predicted time savings that our framework provides in simulation and validating these predictions on complex problems drawn from real computer vision tasks, we are able to reliably predict the performance gain obtained when using a compute cluster to tackle resource intensive computer vision tasks such as 3D point cloud registration.

In summary, this Chapter presents a framework that enables task distribution for computationally demanding problems coupled with a modelling process capable of predicting the available speed benefit of instantiating the distributed implementation. In section 4.2 we first briefly review the task farming problem class. The HPC system

that we make use of experimentally is described in section 4.3.1. We outline our task farming framework and relate it to the BSP model in section 4.3.2. We then introduce performance modelling techniques to facilitate predictions about computational time required for problems formulated under our framework in the remaining parts of section 4.3. Results from simulation experiments that verify our predictive model are given in section 4.4. Section 4.5 details the results of implementing our point set registration algorithm under this task farming framework and results are compared to a sequential implementation of the equivalent problem. Section 4.6 concludes the Chapter with discussion on the advantages that this style of distributed framework brings to the task at hand and some further avenues of exploration are proposed.

4.1.1 *Chapter contributions*

Our contributions in this Chapter can be summarised as follows:

- We introduce a framework for non-independent task farming. The framework allows us to formulate problems by dividing them into many independent parallel tasks that also require some level of communication and synchronisation between tasks before an overall solution to the problem can be obtained.
- As part of this framework we develop a computation-time model capable of predicting overall application completion time for problems that are formulated using the task farming framework that we introduce. This model takes analytical elements from the Bulk Synchronous Parallel (BSP) model [267] and combines these with aspects of simulation based modelling. Providing this simple tool affords a method to reliably predict the time requirements of applications distributed under our framework and therefore evaluate computation-time and solution-quality trade-offs prior to runtime.
- We apply our semi-synchronised task farming framework to a contemporary computer vision problem and report on our experiences of implementing distributed solutions to this problems and explore predicted and experimental speed up available when deploying such implementations on a High Performance Computing (HPC) cluster.

4.2 TASK FARMING

With the advent of multi-core processor architectures and cloud-based platforms, high performance computing is becoming a ubiquitous tool. Distributed compute clusters allow the computing power of heterogeneous (and homogeneous) resources to be utilised to solve large-scale science and engineering problems with increasing uptake in the areas of *e.g.* Medical Imaging, Surgical Robotics, and Pervasive Sensing. One class of problem that has attractive scalability properties, and is therefore often implemented using compute clusters, is task farming (or parameter sweep) applications. A typical characteristic of such applications is that no communication is needed between distributed tasks during overall computation. However interesting large-scale task farming problem instances that do require global communication between tasks sets also exist.

Computational tasks that employ serial code are limited by the total CPU time that they require to execute. When the individual tasks that make up an overall computation are independent of each other it is possible that they run simultaneously (in parallel) on different processors. Using this approach has the potential to greatly reduce the wall-clock time (real-world time elapsed from process start to completion) needed to obtain scientific results. The simple process of distributing separate runs of the same code while varying model parameters or input data is known as *task farming* and makes up an important class of grid computing applications that have been the focus of much initial work [40, 44, 45, 282]. Trivial task farming is a common form of parallelism and relies on the ability to decompose a problem into a number of nearly identical yet independent tasks. Many algorithms are able to fit into such a framework. Each processor (independent node) runs a local copy of the serial code, often with its own input and output files, and no communication is required between these processes. This form of task farming is well suited to exploring large parameter spaces or large independent data sets. On the assumption that all tasks take a similar amount of time to complete, there are no load imbalance issues and linear scaling can often be achieved in relation to the number of processors employed.

We propose a framework called *Semi-Synchronised Task Farming* (SSTF) in order to address problems requiring distributed formulations containing tasks that alternate between independence and synchronisation. In this Chapter we apply this framework

to the previously introduced point cloud registration technique and present a detailed performance analysis to demonstrate framework scalability and benefits obtained.

4.3 SEMI-SYNCHRONISED TASK FARMING

In contrast to the previously outlined task farming, interesting problems that do require some level of communication between tasks during distributed execution also exist. This Chapter details a framework proposed to enable *semi-synchronised task farming* in which an overall computation involves distributing many *sets* of parallel tasks such that all tasks *within a set* are independent yet these tasks must finish before a following task set is able to begin execution. Taking into account communication between tasks has been approached previously with a focus on *e.g.* the scheduling aspects of aperiodically arriving non-independent tasks [2], data staging effects on wide area task farming [85] and cost-time optimisations of task scheduling [41]. Given that we propose to handle global communication between *task sets* with a post task set completion synchronisation step after a round of concurrent computation, components of the Bulk Synchronous Parallel (BSP) model are a suitable basis for our framework. The BSP model is a bridging model originally proposed by [267] and further detail of how to realise our framework and hybrid time prediction model is provided in section 4.3.

Numerical algorithms can often be implemented using either task or data parallelism [97, 127]. Task farming algorithms can be considered a simple subset of task parallel methods that break a problem down into individual segments, such that each problem segment can be solved independently and synchronously on separate compute nodes. The task parallel model typically requires little inter-node communication. Data parallel models conversely share large data sets among multiple compute nodes and then perform similar operations independently on the participating nodes for each element of the data array. Data parallelism therefore typically requires that each processor performs the same task on different pieces of the distributed data. In this way, HPC data parallelism often results in additional communication overhead between nodes and requires high bandwidth and low latency node connectivity. In practice most real parallel computations fall somewhere on a spectrum between task and data parallelism. This is also true of the task farming framework that we introduce (see section 4.3).

Computer vision, like many fields, contains algorithms that are challenged by the size of the data sets worked with, the number of parameters that must be estimated or the requirement of highly accurate results. These requirements often result in computationally expensive algorithms that demand time consuming batch processing. One efficient solution for accelerating these processes involves executing algorithms on a cluster of machines rather than on a single compute node or workstation. Our *semi-synchronised task farming* framework provides a simple form of parallel computation that is able to reduce the wall-clock time required by such computationally expensive tasks that might otherwise take several hours, days or even weeks on a single workstation.

The previously introduced point cloud registration application (Chapter 3) is chosen as a challenging test bed for our distributed framework. Once an algorithm has been formulated under our distributed framework, simple performance modelling is used to accurately predict overall computation time and therefore the likely speed up made possible by employing a distributed implementation over a serial approach.

4.3.1 *HPC experimental implementation*

In this work we make use of the Edinburgh Compute and Data Facility (ECDF) [82] to test the parallel implementations of the computer vision problems that we investigate. The ECDF is a Linux compute cluster that comprises of 130 IBM iDataPlex servers, each server node has two Intel Westmere quad-core processors sharing 24 GB of memory. The system uses Sun Grid Engine [105] (SGE) as a batch queueing system. By tackling computer vision problems through parallel computation with SGE we show that increasing the number of participating processors reduces the wall-clock time required for algorithms implemented under our semi-synchronised task farming framework (see section 4.5 for experimental details). Algorithms are implemented in Matlab and computation times are recorded using the built-in Matlab command `cputime`. We report on the savings due to application speed up in terms of reduced execution time when running our parallel implementations using many processors compared to employing sequential implementations to perform the same tasks. Our parallel implementations make use of the Distributed Computing Engine (DCE) and Distributed Computing Toolbox (DCT) from MathWorks [174]. These products offer a user-friendly method of parallel programming such that master-slave communication between cluster machines

is hidden from the developer, allowing them to focus on domain specific aspects of each problem. Our task farming framework is language independent and we concede that problem instance wall-clock times can likely be reduced further by making use of *e.g.* an alternative compiled language. However the primary focus of the current work is to provide evidence that the proposed framework is able to formulate problems consistently and reduce wall-clock times predictably, compared to the related serial implementations, regardless of the language used. We leave a study of time critical applications benefiting from *e.g.* compiled languages like C/C++ to future work.

4.3.2 *The Bulk Synchronous Parallel model*

The BSP model is a bridging model originally proposed by [267]. It is a style of parallel programming developed for general purpose parallelism, that is parallelism across all application areas and a wide range of architectures [175]. Intended to be employed for distributed-memory computing, the original model assumes a BSP machine consists of p identical processors. The related semi-synchronised farming framework we propose (section 4.3.3) does not strictly enforce a homogeneous resource requirement in comparison. This enables our experimental setup, using IBM iDataPlex servers, to contain similar but not necessarily identical nodes. In accordance with the original BSP model we do assume homogeneous resources during our theoretical performance modelling for simplicity and we therefore leave a heterogeneous performance modelling treatment to future work. In the original BSP model, each processor has access to its own local memory and processors can typically communicate with each other through an all-to-all network. In our work we make the simplifying assumption that processes only contribute information to a global decision making process at the end of each *set of tasks* and therefore do not need to communicate with each other directly. A BSP algorithm consists of an arbitrary number of *supersteps*. During supersteps, no communication between processors may occur and all processes, upon completing their current task must then wait at a *barrier*. Once all processes complete their current task a *barrier synchronisation* step occurs and then the next round of tasks (superstep) can begin. In this fashion a BSP computation proceeds in a series of global supersteps and we utilise these supersteps to model *sets of* parallel distributed tasks in our framework. To summarise, a superstep typically consists of three components:

1. Concurrent computation: computation takes place on each of the participating processors p. Processors only make use of data stored in the local processor memory. Here we call each independent process a *task*. These *tasks* occur asynchronously of each other.

2. Communication: Processors exchange data between each other. Our framework makes the simplifying assumption that *tasks* do not need to exchange data with each other individually yet the result of each local computation contributes to the following Barrier synchronisation step (global decision making). This assumption holds for the application that we investigate (see section 4.5).

3. Barrier synchronisation: When each *task* reaches this point (the barrier), it must wait until all other *tasks* have finished their required processing. Once all *tasks* have completed, we make a set of global decisions before the next superstep may begin (the next round of concurrent computation and so on).

4.3.3 *Theoretical framework*

As noted, our strategy involves global communication between *task sets* during a post task-set-completion synchronisation step following a round of concurrent computation. The components and fundamental properties of the Bulk Synchronous Parallel (BSP) model provide a suitable basis for this framework. Namely moving from a sequential implementation to describe the use of parallelism with a BSP model requires only a bare minimum of extra information be supplied. BSP models are independent of target architecture, making a task farming framework based on BSP portable between distributed architectures. Finally the performance of a program distributed using a BSP inspired framework is predictable if a few simple parameters from the target program can be provided (*e.g.* task-length distribution parameters). Towards this goal, we combine the standard analytical elements from the BSP model with components of simulation based modelling leading to a novel hybrid performance modelling technique capable of predicting the runtime of algorithms implemented with our framework.

We solve large scale problems by sharing large data sets among multiple processors. The *Semi-Synchronised Task Farming* framework, in consonance with a task parallelism model, involves only little inter-node communication between tasks running in parallel.

However, similar to data parallelism models, the framework allows us to split these large data sets between compute nodes and perform independent calculations on participating processors in parallel. As the calculations *within each task* are independent, no information needs to be exchanged between nodes during task runtime and sharing of results is postponed until all tasks in a set have completed. As discussed, once a set of tasks has been completed we are able to collate results and use this information to make decisions relating to how the following round of tasks should be formulated. The outputs from the final round of tasks are combined to provide the global program output. This framework is formally defined in the following pseudocode and Figure 40 depicts the process in diagrammatic form.

Let:
$\{I_i^{[t]}\}_{i=1}^{N_t}$ be the set of N_t input tasks at superstep t
$\{O_i^{[t]}\}_{i=1}^{N_t}$ be the set of N_t outputs gained from the tasks completed at superstep t
Input: N_0 tasks at superstep $t = 0$
begin
  terminate := 0
  while (NOT terminate)
    parallel for $i \in N_t$
      $O_i^{[t]} = \text{process}(I_i^{[t]})$
    end
    $\{I_{i=1}^{[t+1]}\}_{i=1}^{N_{t+1}} = \text{recompute\_inputs}(\{I_{i=1}^{[t]}\}_{i=1}^{N_t}\ ,\ \{O_{i=1}^{[t]}\}_{i=1}^{N_t})$
    $\text{terminate} \stackrel{?}{=} \text{test\_termination\_criteria}(\{O_i^{[t]}\}_{i=1}^{N_t})$
    $t = t + 1$
  end
  $last = t$
  $R = \text{combine\_outputs}(\{O_i^{[last]}\}_{i=1}^{N_{last}})$
end
Output: R

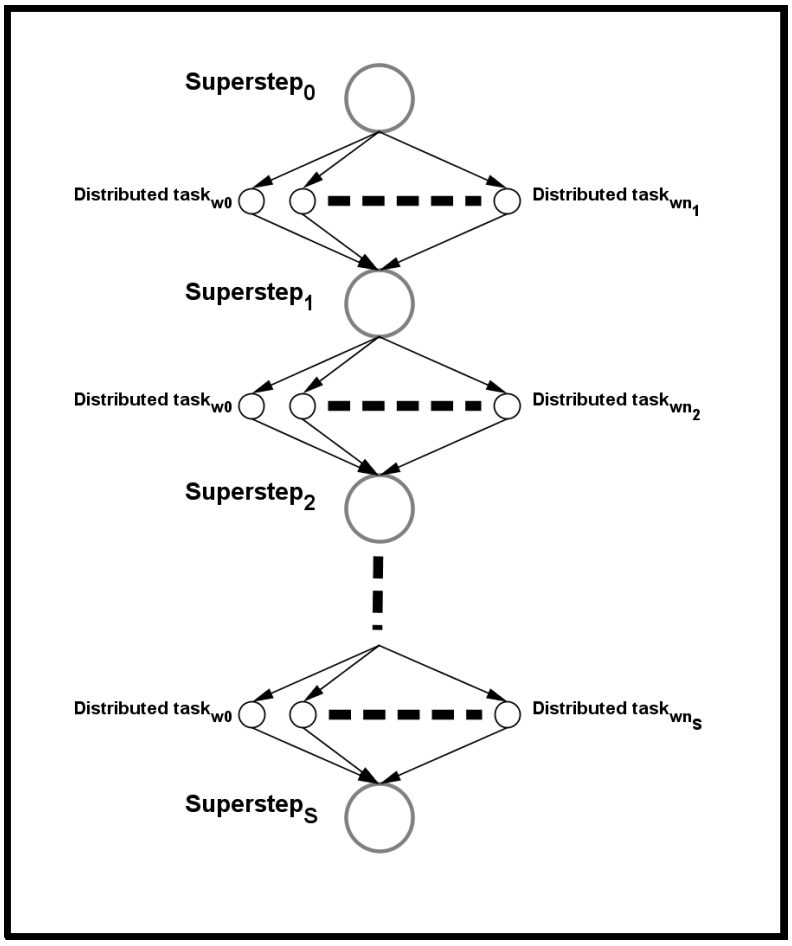
Each task in a task set is distributed to an individual processor and *tasks* following each *superstep* are not regarded as having a particular linear order (from left to right or otherwise) and may be mapped to processors in any way. The provided pseudocode dictates re-computation of inputs prior to testing termination criterion. It is noted that

this may result in slight inefficiency in practice however as is commonly the case (*e.g.* the distributed application considered in this work), the re-computation of inputs is of negligible cost. Cases involving input re-computation with non-negligible cost can be efficiently addressed using straightforward modifications.

The advantage of adding the BSP synchronisation step between task sets allows all tasks in a set the opportunity to collate and communicate information resulting from the completion of their collective execution. The collective results of a task set can influence decisions involving the form, model parameters and possibly the number of tasks making up the following task set input. Once formulated, the following set of tasks can be distributed to the participating processors. It is this process of dispatching multiple rounds of parallel independent tasks, where task formulation may be influenced by information from previous task set results, that we call *Semi-Synchronised Task Farming*. This approach allows us to find distributed solutions to non-trivial problems that require a level of communication between nodes during overall computation while retaining much of the simplicity of the standard task farming model. If all tasks *within a task set* take a similar amount of time to complete then it allows for simple modelling and task distribution. If however tasks exhibit completion times with high variance, then a smart scheduler (such as SGE [105]) can still be used efficiently to ensure that load balancing is not problematic for our framework. The wall-clock time, now related to both the number of task sets and the number of available processors, is much improved over serial implementations.

The synchronisation aspect allows us to solve problem decompositions that require a level of inter-node communication while retaining the main advantages of a standard task farming approach such as ease of implementation, level of achievable efficiency (on the assumption that individual tasks in a set require similar time to complete) and, given that existing serial code can often be used with minimal modification, users can produce solutions without requiring detailed knowledge of *e.g.* MPI techniques. We do however note that if tasks take widely different amounts of execution time then the total wall-clock time of a task set is no better than the slowest process. Being more precise than this is hard because the wall-clock time of a task set also depends on the number of CPUs (and tasks) taking part in the computation (see section 4.3.6.2 for further discussion of this point).

Figure 40: The *Semi-Synchronised Task Farming* framework. Light grey *superstep* nodes indicate task synchronisation and collective global decisions based on information obtained from the previous set of distributed *tasks*. These decision points influence the input data, form (and possibly the number) of the following set of distributed *tasks*. See text for further detail.



4.3.4 *Simulation and analytical hybrid performance modelling*

We undertake simple performance modelling to evaluate the distributed job submission behaviour on a CPU cluster allowing prediction of the run time performance of algorithms realised with our framework. Performance modelling of distributed systems enables an understanding of code and machine behaviour and can be broadly split into two categories; analytical modelling and simulation based techniques. Analytical models are typically developed through the manual inspection of source code and subsequent formulation of critical path execution time. This approach usually involves the implementation of a modelling framework (*e.g.* LoPC [100]) to reduce the work required by the performance modeller. Analytical approaches are effective yet often require manual analysis of source code necessitating knowledge of the task domain, implementation languages and communication paradigms.

Here we follow a coarse grained alternative approach of simulation based performance modelling. Many simulation tools exist to support this form of performance modelling (*e.g.* the DIMEMAS project [155]). Such tools often involve replaying the code to be modelled instruction-by-instruction and the related use of machine resources can then be gathered by the simulator. More recent work such as the WARPP tool kit [115, 116] make use of larger computational events (as opposed to instruction based simulation) improving simulator scalability. Here we take a similar approach; instead of using single application instructions we model coarse grained computational blocks. We choose a coarse level of granularity by defining a computational block as one distributed task in our framework. We then obtain run times for these computational blocks through traditional code profiling. An additional advantage of this coarse-grained simulation is that hybrid models (combining analytical and simulation-based approaches) can be built. By combining these coarse-grained computational events with an analytical model typical of the Bulk Synchronous Parallel (BSP) [267] model we obtain a straightforward hybrid model capable of predicting application run-time for the algorithms that we implement using our task farming framework.

4.3.5 *BSP cost in relation to task farming*

The cost of an algorithm represented by the BSP model is defined as follows. The cost of each superstep is determined by the sum of three terms; the cost of the longest running local *task* w_i, the global communication cost g per message between processors where the number of messages sent or received by *task* i is h_i and the cost of the barrier synchronisation at the end of each superstep is l (which may be negligible and therefore the term is dropped).
The cost of one superstep for p processors is therefore:

$$\max_{i=1}^{p}(w_i) + \max_{i=1}^{p}(h_i g) + l \tag{20}$$

We make standard simplifying assumptions that we have $\geq p$ homogeneous processors and that *tasks* do not need to exchange data with each other individually or with the master node during each superstep thus ensuring that $h_i = 0$ for all i. We assume homogeneous processors for simplicity during our cost treatment but note that in the current landscape of computation, heterogeneous resources are also common. Although our framework is applicable to heterogeneous resources in practice, we leave a theoretical treatment of heterogeneous processor cost to future work (see section 4.4 for related discussion of this point). It is common for Equation 20 to be written as $w + hg + l$ where w and h are maxima and with our simplification this reduces further to $w + l$. The cost of the algorithm then, is the sum of the costs of each superstep where S is the number of supersteps required.

$$W + Hg + Sl = \sum_{s=1}^{S} w_s + 0 + Sl \tag{21}$$

4.3.6 *Empirical simulation and modelling*

We simulate total parallel algorithm execution times by firstly generating random trials to simulate individual distributed *task* timings. To simulate a real-world task set, we generate trials from a Gaussian distribution parametrised by the mean time required in practice for a single distributed task to complete and add these to the time cost of

barrier synchronisation. Task timing distribution parameters are found through code profiling and making use of the Matlab function `cputime`. We assert that this is a reasonable method to simulate task timings as the task farming applications that we investigate all distribute sets of similar length tasks during each superstep. By specifying or observing the number of *supersteps* required for a given real world computation and the number of distributed tasks required in each *superstep*, we are able to approximate the total time required by the parallel algorithm as:

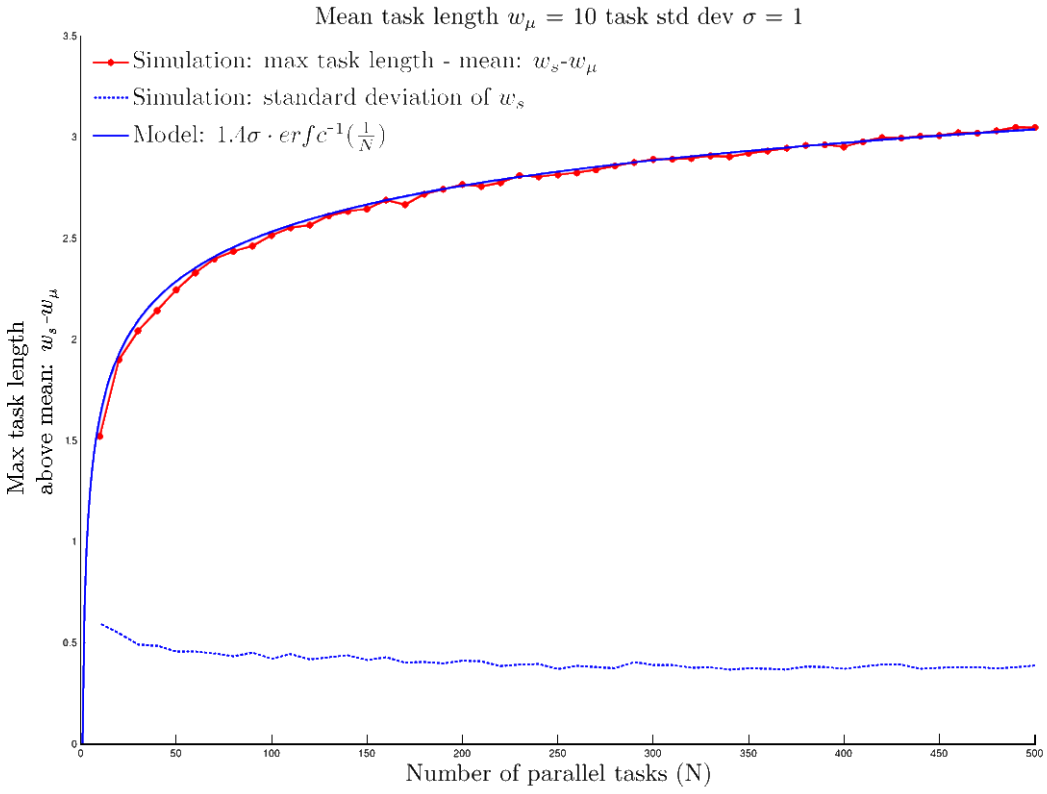
$$\sum_{s=1}^{S} w_s + Sl \tag{22}$$

where w_s is the longest running local task in superstep s, barrier synchronisation time cost is l and the total number of supersteps is S. In practice we run this simulation over many trials and look at the mean result for an algorithm that requires N_s distributed tasks during each superstep.

4.3.6.1 *Limitless CPU node model*

As a simple example we take a mean task length of $w_\mu = 10$ time units and a task length standard deviation of $\sigma = 1$, and simulate an application making use of only a single superstep. We find that, using the additional assumption of limitless computational nodes, as we increase the number of distributed tasks required in the superstep the difference between the longest task length w_s and the mean task length w_μ grows sub-linearly with the number of submitted distributed tasks N (using 1000 trials per data point in Figure 41). From this simple example we are able to conclude that, not taking into account limited computational resources, if we have an application that benefits from increasing the number of distributed tasks during a superstep (*e.g.* by an order of magnitude - see for example section 4.5), we can expect improved results for only a small increase in predicted wall-clock time cost.

Figure 41: Predicted difference between maximum distributed task time and mean task time $w_s - w_\mu$, where $w_\mu = 10$, $\sigma = 1$ for an algorithm distributing N tasks in one superstep. Simulation values obtained using 1000 trials per data point.



We can fit this simulated computation time accurately using the standard inverse complementary error function. The complementary error function *erfc* (also known as the Gaussian error function) provides us with an accurate predictor for the maximum job length w_s increment over the mean job length w_μ, in relation to the number of submitted jobs, that we are likely to observe assuming that the true job length distribution resembles a Gaussian distribution. The *erfc* function is often used in statistical analysis to predict behaviour of any sample with respect to the population mean. Here we fit our simulation data by applying the inverse *erfc* to $\left(\frac{1}{N_s}\right)$, where N_s is the number of submitted tasks in superstep s (see Figure 41). The error function *erf* is defined as:

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

Then the complementary error function, denoted *erfc* and its inverse $erfc^{-1}$ are defined as:

$$\mathrm{erfc}(x) = 1 - \mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt$$

$$\mathrm{erfc}^{-1}(1 - x) = \mathrm{erf}^{-1}(x)$$

The model that empirically fits the simulation for mean task length w_μ, with standard deviation σ distributing N_s tasks in parallel, lets us predict the maximum task time w_s for superstep s as:

$$w_s = w_\mu + \left(1.4\sigma \cdot \mathrm{erfc}^{-1}(\frac{1}{N_s})\right) \tag{23}$$

The scalar 1.4 is needed to fit our empirical data. We hypothesise that the true scalar value providing the best fit to our empirical curve here is $\sqrt{2}$ but we leave investigation of this to future work. In Figure 41 we use $w_\mu = 10$ and $\sigma = 1$ and simulate for various task set sizes N_s. If computational resources are not a limiting factor, then once we know the number of distributed tasks N_s required per superstep, and have estimates for w_μ and σ we are able to approximate the expected time w_s required for a single superstep of a given algorithm and, given the number of supersteps, the expected time required for the entire algorithm. This model is valid in cases where the number of available parallel worker processors is equal to or exceeds the number of tasks required per superstep. We have access to 130 iDataPlex servers with multiple CPUs, however

in many practical applications this requirement will not hold (the number of tasks per superstep will exceed available participating worker nodes) therefore we also consider a finite CPU model in the following section.

4.3.6.2 *Finite CPU node model*

The previous simulation model does not take into account CPU worker node limits. In this section additional simulations are performed to explore the effect of capping the number of available CPU nodes K in relation to the number of submitted distributed tasks per superstep N_s. This allows us to fit a model that reflects our real distributed system pragmatically. In this case, we assume that $N_s > K$ and therefore each CPU node is responsible for the computation of a number of tasks in sequence in order to complete a superstep. In our task farming framework under SGE, when a CPU worker node completes the computation of the current task then the next task from the set still waiting to be processed will be assigned to the finished core such that each core is continually utilised until all tasks have been processed. For each simulation trial, the maximum cumulative CPU computation time used by a worker node during a superstep; CPU_s must now be found. This value is the maximal sum of task computation times assigned to an individual CPU. From this max cumulative computation time found during a superstep, we subtract $w_\mu \cdot \left(\frac{N_s}{K}\right)$ where w_μ is the mean task length, N_s is the number of parallel tasks making up the superstep and K is the number of participating processors. This effectively subtracts the mean amount of work we expect a CPU to perform per superstep. This mean amount of work per CPU is denoted $CPU_\mu = w_\mu \cdot \left(\frac{N_s}{K}\right)$. The resulting difference tells us how much more work, than the mean cumulative work, we expect the node assigned the most work to carry out. As a result, CPU_s provides the time we expect the full superstep s to take to complete.

The final point above holds because all CPU worker nodes must be allowed to finish their assigned cumulative task computation before it is possible to synchronise and conclude a superstep s. When accounting for a finite set of CPU worker nodes we therefore model the time it takes to complete a superstep s as the longest cumulative

CPU computation time CPU_s. When accounting for a fixed number of worker nodes K, the model that we find (approximately) empirically fits the simulation data is:

$$CPU_s = \begin{cases} w_\mu \cdot \left(\frac{N_s - \text{mod}(N_s,K)}{K}\right) + w_\mu & \text{if mod}(N_s,K) \neq 0 \\ w_\mu \cdot \left(\frac{N_s}{K}\right) + 1.4\sigma \cdot \text{erfc}^{-1}(\frac{1}{N_s}) & \text{if mod}(N_s,K) = 0 \end{cases} \tag{24}$$

where

$$CPU_\mu = w_\mu \cdot \left(\frac{N_s}{K}\right)$$

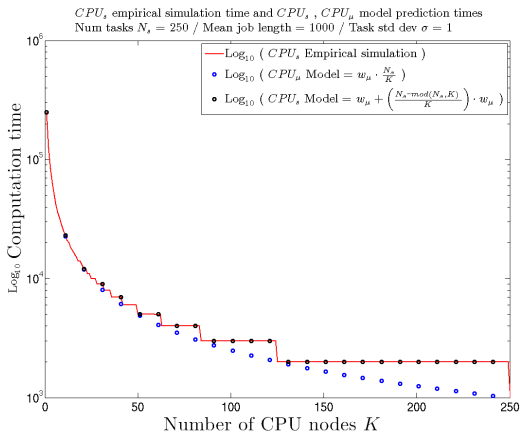
We model CPU_s as the mean computational work done at each worker, CPU_μ plus some additional work that must be carried out by the CPU that has performed the most work in the current superstep. We model this additional work in the following way: when we consider a finite set K of CPU worker nodes, the difference between the longest cumulative CPU computation time CPU_s and the mean cumulative CPU computation time CPU_μ is primarily influenced by: 1) how evenly the number of distributed tasks N_s are distributed to the number of participating CPU nodes K and 2) the mean task length w_μ. Advanced task farm models (*e.g.* [203]) employ various strategies dictating how tasks should be distributed to workers. Here we take the simple approach that, on the assumption that tasks belonging to a task set have similar length, each task still waiting to be processed will be assigned in turn to the CPU worker node that finishes its current computational work load first. A consequence of this is that if the total number of distributed tasks N_s required by the superstep is exactly divisible by the number of participating CPU nodes K (*i.e.* $\text{mod}(N_s,K) = 0$) then, excluding cases involving extremely high task length variance σ^2 in relation to w_μ, each CPU will receive an identical number of tasks and therefore the difference between the longest cumulative CPU computation time CPU_s and the mean time CPU_μ will be small and only influenced by the number of tasks N_s and the task length variance σ^2 in a similar fashion to the limitless worker node model. In such cases this small difference is once again accounted for using the $erfc^{-1}$ function as before (see Figure 41 and Equation 23). If, contrarily, the number of tasks N_s divided by the number of participating CPU nodes K leaves a remaining number of tasks that is small in relation to K (*i.e.* $\text{mod}(N_s,K) \ll K$) then, again assuming moderate task length variance σ^2 in relation to w_μ, the CPU node completing the most computational work will contain one more task than

$\lfloor\left(\frac{N_s}{K}\right)\rfloor$. We account for this additional task in our model by adding the mean task length w_μ (our additional task) to the mean cumulative work done, adjusted by the number of CPU worker nodes that are assigned an additional task such that they must complete $\lfloor\left(\frac{N_s}{K}\right)\rfloor + 1$ tasks in total.

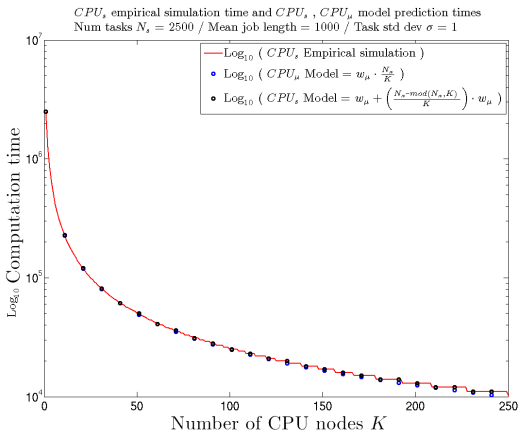
This models the fact that the difference between CPU_s and CPU_μ will be greater when fewer worker nodes are assigned $\lfloor\left(\frac{N_s}{K}\right)\rfloor + 1$ tasks to complete since the true mean work done per CPU will be close to $w_\mu \cdot \lfloor\left(\frac{N_s}{K}\right)\rfloor$ when many nodes are completing only $\lfloor\left(\frac{N_s}{K}\right)\rfloor$ tasks. The difference between CPU_s and CPU_μ is therefore essentially linear in mean task length w_μ once N_s, K and σ are known. Intuitively, if $\text{mod}(N_s, K)$ is low but non-zero *e.g.* equal to one, then the single CPU that is assigned this extra task will be required to complete almost exactly one extra task length of work in comparison to the mean amount of work $CPU_\mu \approx w_\mu \cdot \lfloor\left(\frac{N_s}{K}\right)\rfloor$. As $\text{mod}(N_s, K)$ grows, the value representing the mean amount of work done per CPU is adjusted accordingly. The special case where $\text{mod}(N_s, K) = 0$ we expect, as discussed previously, only adds a constant amount of excess work above the mean for large N_s similar to the case explored previously using an unbounded K (see section 4.3.6.1).

We validate this model using empirical simulation data for various K and task set sizes N_s. A sample of these simulation and model prediction results, exploring simulated and predicted times for $K \in \{1 \ldots 250\}$ are found in Figures 42a - 42d for the case where we fix $w_\mu = 1000$. Empirical simulation data point values are averaged over 1000 trials. In the Figures 42a - 42d we show simulations distributing N_s tasks over a single superstep with a mean task length of $w_\mu = 1000$. As might be expected, as N_s is increased to the point that $K \ll N_s$ (subfigure 42b) the difference between mean and maximum work carried out by CPUs converges to zero. More interestingly, as task length standard deviation σ is increased, in relation to mean task length w_μ, (subfigures 42c and 42d) minor discrepancy emerges between empirical simulation and our CPU_s model. These differences remain negligible for standard deviation magnitudes similar to those found experimentally when measuring real-world task time lengths (*c.f.* section 4.4 and Figure 44). Our CPU_s model exhibits an acceptable level of robustness to levels of task length variation found in practice. Empirical simulation (red line) data points are averaged over 1000 trials.

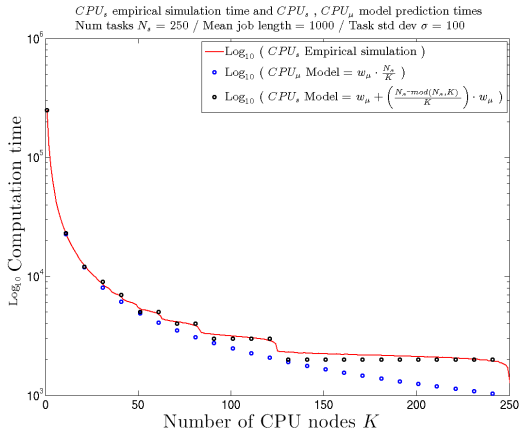
Additionally, Figure 43 illustrates the difference between our model predictions CPU_s and empirical simulation for various K. For the number of CPUs K that we are likely



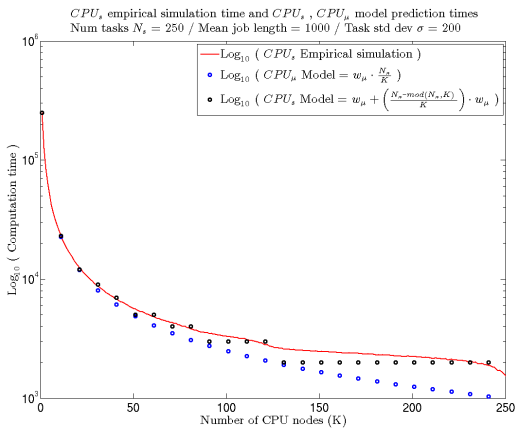
(a) N_s=250 , σ=1 , w_μ=1000 , $K \in \{1,\ldots,250\}$



(b) N_s=2500 , σ=1 , w_μ=1000 , $K \in \{1,\ldots,250\}$



(c) N_s=250 , σ=100 , w_μ=1000 , $K \in \{1,\ldots,250\}$



(d) N_s=250 , σ=200 , w_μ=1000 , $K \in \{1,\ldots,250\}$

Figure 42: CPU_μ (blue 'o' shown for every 10th K) provides a simple model of the mean work we expect processors to carry out in terms of total (log-scale) computation time units when distributing tasks over K processors. We show, using empirical simulation (red line), how the longest CPU queue (maximum work done) CPU_s deviates from this value in practice in relation to number of tasks N_s and processors K. Model prediction of this maximum work carried out by a CPU: 'CPU_s Model' (black 'o' plotted for every 10th K value) exhibits how accounting for this extra work improves the accuracy of the predicted overall completion time (see text for detail).

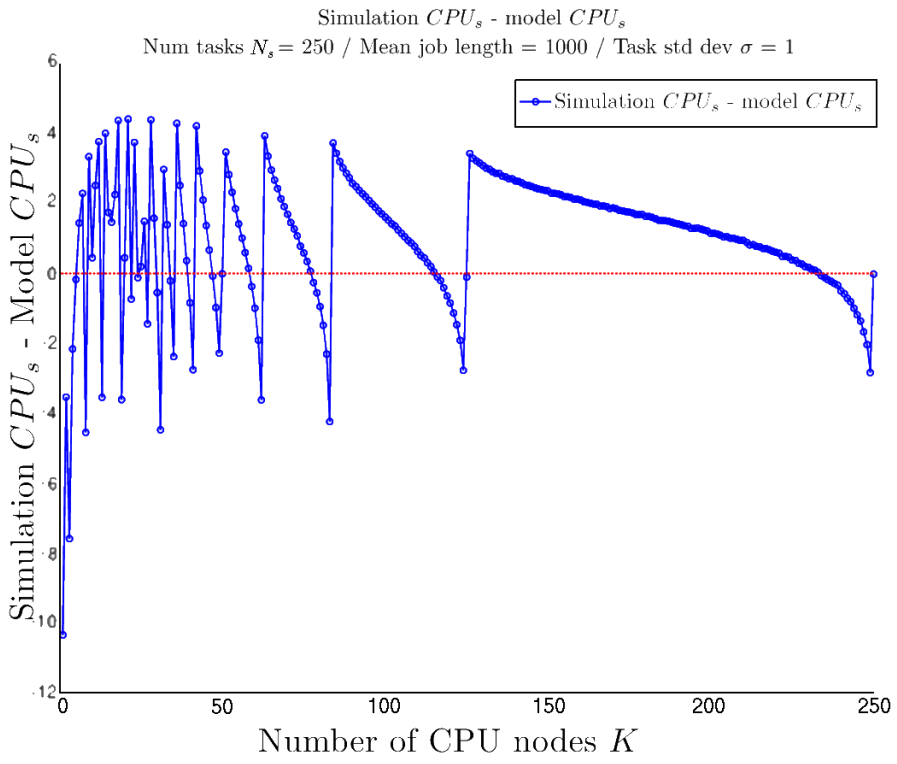


Figure 43: Model prediction error and empirical simulation for each value of $K \in \{1..250\}$. We exhibit model prediction error of < 10 time units (Y-axis) when using $\sigma = 1$ and a mean job length $w_\mu = 1000$ time units for each value of K explored. Our prediction makes small periodic errors but this error reduces further as K increases. For the number of CPUs that we make use of in practice (*e.g.* > 20) we see an overall computation time prediction error of < 4 time units when using $w_\mu = 1000$ time units.

to make use of in practice (*e.g.* > 20) computation time model prediction error is $\leq 1\%$ when compared to empirical simulation (we concede that this error size increases when predicting real-world application runtime. See section 4.4 for additional detail).

4.4 SSTF MODELLING FOR SGE DISTRIBUTED APPLICATIONS

In this section, we use our SSTF model (introduced in sections 4.3.6.1 and 4.3.6.2) to predict the expected run time of real-world applications that we distribute on our SGE cluster under the task farming framework. We present results from job submission under real network and Grid Engine loading conditions and compare measured runtime results with predictions to test the validity of the models developed in section 4.3.

Various application configurations are submitted to the SGE cluster that involve distributing $N_s = 20$, 40 and 100 tasks during each superstep in applications making use of $S = 5$, 10 and 30 supersteps. The application that we utilise for testing our model contains parallel tasks with cost durations of comparable length by design. Further details regarding the distributed version of the point registration algorithm (utilised here experimentally) are given in section 4.5.

To calculate true overall application time cost we record individual parallel task run times and are, therefore, able to find the longest running (highest cost) task within each superstep. We then sum the times required for the longest running task w_s in each superstep s such that:

$$\sum_{s=1}^{S} w_s + Sl$$

provides the total time needed to execute the parallel application in practice (*c.f.* equation 22), assuming that all tasks within a superstep are able to run in parallel. With regard to the sample application that we investigate, it is found that the time cost for *barrier synchronisation* steps l are negligible in practice and therefore we neglect these in the runtime calculation. Although barrier synchronisation is negligible in the sample application investigated here, we note that this is certainly not always the case and we, therefore, choose not to oversimplify the model.

Repeated trials ($n = 10$) are performed for each application (N_s, S) configuration tested. Detail of a configuration distributing $N_s = 20$ tasks during each of $S = 10$ supersteps is now provided as an example. In this example real-world runtime measurement results in a total cost of $\sum_{s=1}^{10} w_s = 123.06$ minutes of parallel computation time with a mean measured task length of $w_\mu = 462.9$ seconds (~ 8 minutes), and a task length standard deviation of $\sigma = 107.13$ seconds. These values are obtained by averaging across the $n = 10$ experimental trials. The recorded individual task times, across all supersteps from one $(N_s = 20, S = 10)$ trial, are shown in Figure 44. Individual runtime costs are obtained by profiling the application through the use of the Matlab function `cputime`.

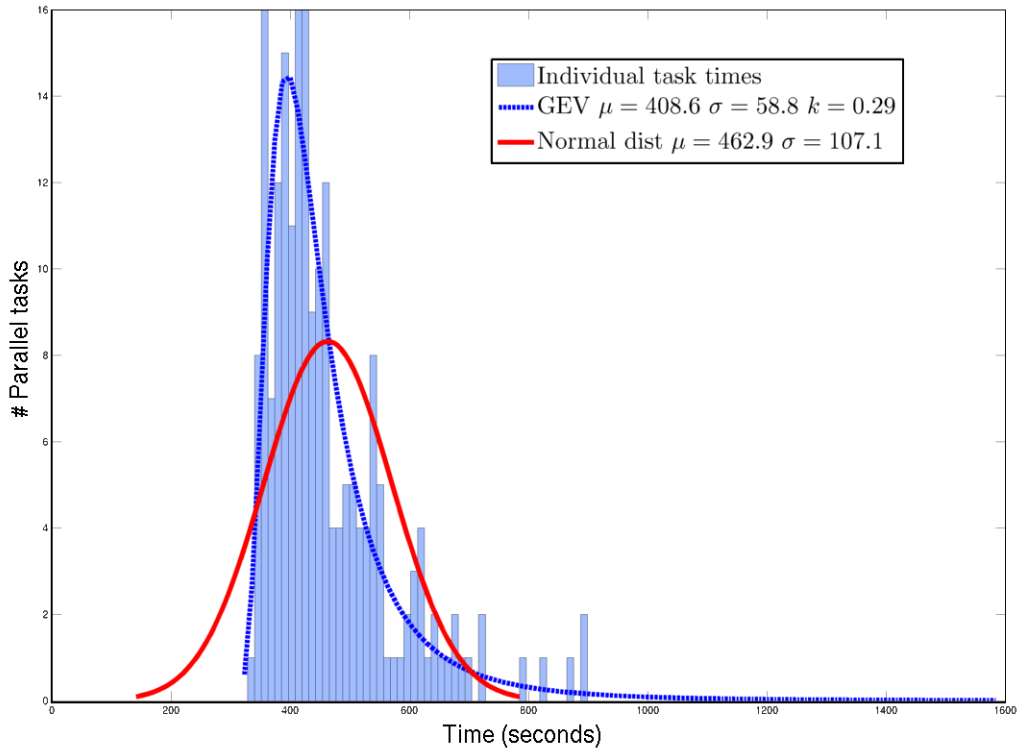


Figure 44: Individual parallel task timings recorded experimentally across 10 supersteps from one trial with Gaussian, GEV best fit models used to explore the parallel task set timing distribution assumptions.

Using the distributed task model that we introduce in Equation 24, and assuming that we have sufficient participating processors K to accommodate $N_s = 20$ tasks in parallel, we predict the maximum work performed by a single processor in a superstep to be $CPU_s = 669.86$ seconds for this example (an underestimation, the mean of the maximum values found in practice, across $n = 10$ trials, for this configuration is 738.37 seconds of CPU time). Using $S = 10$ supersteps the total runtime predicted by our model for this experiment is therefore 111.6 minutes. This results in an underestimation of the true mean total measured time by 11.4 minutes ($\sim 10\%$) for this (N_s, S) configuration. This underestimation may be explained by the slightly non-Gaussian distribution of task length observed (*e.g.* Figure 44). Examining the real-world run times of the distributed tasks highlights a slightly heavy-tailed distribution for the particular application employed in this experiment. This typically results in several long runtime outliers that contribute to the total runtime cost using our overall runtime calculation method. For expository purposes we also fit a GEV (Generalised Extreme Value) model to the data here, providing a reasonable fit (*i.e.* resulting in a slightly lower Bayesian Information Criterion (BIC) value of 2343.39 compared to the Gaussian BIC of 2446.78 for this data set). Future work could re-examine our hybrid model using *e.g.* a GEV distribution in place of our current Gaussian timing model to predict run times in cases where this provides a better fit to the independent task length distribution. We also note that one potential route towards accounting for heterogeneous participating processors p during runtime prediction would involve making use of mixture distributions (*e.g.* a mixed GEV distribution). We leave more sophisticated task time distribution fitting to future work.

The (N_s, S) configurations investigated and all predicted and measured job completion times are summarised in Tables 3 and 4. In Table 4 we present measured and predicted **overall computation time** and note that the difference between measured time and our model prediction is always within 11% of the measured value. Additionally, in Figure 45 we show experimentally obtained individual task run times recorded when distributing 100 tasks in parallel across $S = 5$ supersteps.

Our approximate model provides a simple yet moderately accurate method for predicting the amount of computational work required by applications formulated under our task farming framework and distributed to the Sun Grid Engine or some other queue based cluster system. For completeness, we contrast the computational time required

to mean wall-clock time used by the cluster in practice. We note in general wall-clock time is significantly larger than required computational time however we find that in practice wall-clock time is subject to high variance between trials as we have little control over wall-clock time in a multi-user cluster environment. This is mainly due to resources available and the queueing aspect of sharing the SGE cluster with other users. By additionally including Sun Grid Engine queueing (non-working) time, mean wall-clock time for the application run in the provided example was 173.46 minutes (non-working time is attributed to sharing the SGE cluster with other users).

Table 3: Parameter sets used for four different sets of distributed application experiments varying the number of distributed tasks (N_s) and supersteps (S).

	# CPU nodes (K)	Tasks per superstep (N_s)	Supersteps (S)
Model prediction (eq. 24)	20	20	10
Measured timing set 1	20	20	10
Model prediction (eq. 24)	20	20	30
Measured timing set 2	20	20	30
Model prediction (eq. 24)	20	40	05
Measured timing set 3	20	40	05
Model prediction (eq. 24)	20	100	05
Measured timing set 4	20	100	05

Table 4: Distributed application measured timing results and BSP model predictions for four sets of distributed tasks with rows corresponding to Table 3. We obtain the predicted overall computation time by taking the product of the predicted w_s and the number of supersteps (S). The difference between our overall computation time model predictions and measured results are always within 11% of the true value.

	True w_μ (sec)	Task time σ	Predicted w_s (eq. 24) and Measured w_s (sec)	**Overall computation time** (min)	Wall-clock time (min)
Model prediction (eq. 24)	N/A	N/A	(462.0 + 207.86)=669.86	(669.86 sec ·10)=111.6	N/A
Measured timing set 1	462.0	107.13	738.37	123.06	173.46
Model prediction (eq. 24)	N/A	N/A	(348.17 + 168.02)=516.19	(516.19 sec ·30)=258.1	N/A
Measured timing set 2	348.17	86.60	740.0	287.4	434.08
Model prediction (eq. 24)	N/A	N/A	(57.1 + 19.8)=76.9	(76.8 sec ·5)=6.40	N/A
Measured timing set 3	57.1	8.95	91.3	6.89	41.3
Model prediction (eq. 24)	N/A	N/A	(214.4 + 96.46)=310.86	(310.86 sec ·5)=25.9	N/A
Measured timing set 4	214.4	37.83	353.6	27.3	133.0

4.5 DISTRIBUTING MULTI-VIEW POINT CLOUD REGISTRATION

As discussed our approach to multi-view registration can be considered a computationally demanding computer vision problem in cases where many viewpoints are considered. Here computational issues are addressed by proposing an implementation of our registration methodology using the introduced Semi-Synchronised Task Farming framework. We focus on the previously introduced registration task as it provides an example application that is able to benefit from performing many tasks in parallel yet also requires a form of communication between rounds of parallel tasks (supersteps). As described previously, these parallel task sets and synchronisation steps make up a larger computational process. The example application that we study has the following properties that are common to many computationally demanding applications:

- Large input data set. Our input data (*e.g.* point sets containing hundreds of thousands of points) are large relative to the number of model parameters (*e.g.* adaptive kernel density bandwidth parameters h and kernel class) and control options (*e.g.* number of required supersteps and optimiser iteration limits) that dictate the data processing procedures.

- Large number of tasks. The number of tasks N that make up the overall computational process is large (*e.g.* 100 viewpoints (represented by point clouds) taking part in 10 iterations of simultaneous view alignment will result in 1000 tasks). The total number of tasks may also not be known in advance for some applications. Each application launches sets of tasks that are processed in parallel. All tasks in a synchronised superstep must complete before the following round of tasks (superstep) can begin. Task parameters are defined by fixed model parameters and potentially by information resulting from the completion of previous task sets.

- Task independence. Each task is defined by model parameters, the global input data and potentially the task set results from the previous superstep. For tasks that are contained in *the same superstep*, no dependencies exist between superstep task members.

As noted previously, registration can be considered one of the crucial stages of reconstructing 3D object models using information obtained from range images captured

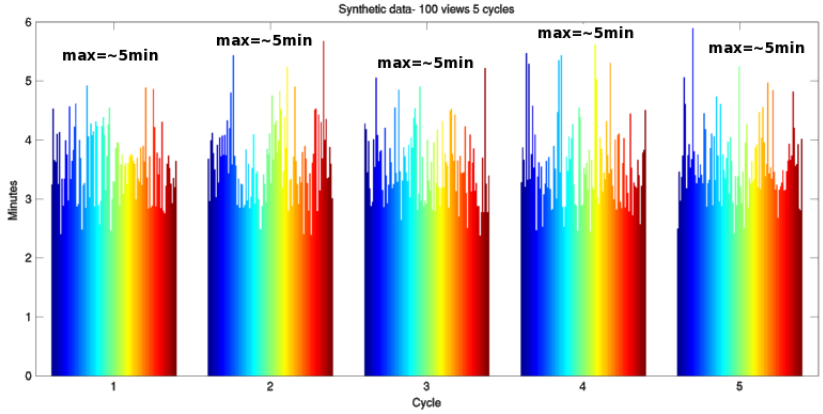
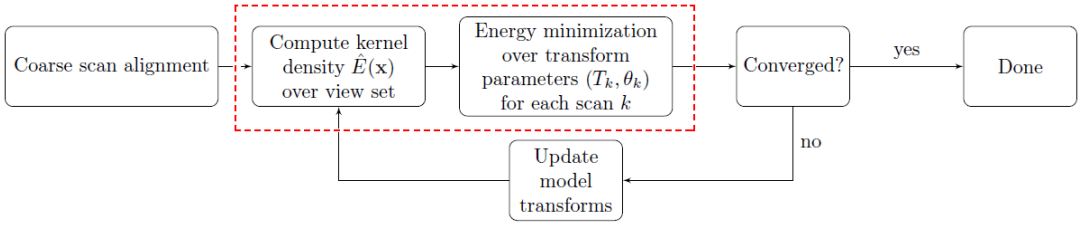


Figure 45: An example distributed task set containing 100 viewpoints over 5 transform cycles. Under the assumption that we have enough cores to run the tasks for each cycle in parallel, the entire procedure is completable in $\sim$ 25 minutes while a serial implementation performing a similar optimisation would take $>$ 40 hours. See Table 4 for further details.

Figure 46: Our multi-view registration method. Stages of the algorithm within the dashed line (red) area are distributed to our cluster in parallel.



from differing object viewpoints. The generalised problem of globally aligning *multiple* partial object surfaces is a difficult task that remains a fundamental part of extracting complete models from multiple 3D surface measurements. The framework outlined in this Chapter allows us to process large numbers of range images per object reconstruction in feasible time frames whilst retaining the accurate, high quality view alignment results typical of simultaneous registration approaches.

Chapter 3 provided detail on our registration strategy (see *e.g.* the pseudocode provided in section 3.4 and Figure 16). Since range viewpoints are aligned in parallel, using our semi-synchronised framework, we are able to accommodate many view sets

for smaller incremental wall-clock time increase than typical serial solutions. Utilising many object viewpoints, for the task of object reconstruction, affords benefits over sparse sets of views such as better object surface coverage, hole filling and reconstructed object detail improvement.

For N viewpoint data sets we define N independent parallel tasks in each superstep and in each of these tasks, as detailed in Chapter 3, we use the current pose of the remaining $N-1$ scans for the purpose of computing a surface estimate and a related energy function. We allow the final, active scan to move in the transform space by searching for optimal pose parameters. Each parallel task assigns a different viewpoint as the active scan. Independently evaluating the position of each moving scan in relation to the inferred surface and therefore minimising our energy function brings the active view into better alignment. Since viewpoint position evaluation is the most computationally expensive part of the procedure, if many viewpoints are made use of then parallelisation of this step typically affords a large time saving. After this minimisation has taken place for each viewpoint in parallel, we have N sets of optimal rigid transform parameters; typically three translation (θ_x, θ_y, θ_z) and three rotation (θ_α, θ_β, θ_γ) parameters that bring each view into alignment with the estimated surface (and therefore the other views). Once each independent task has found a set of rigid transform parameters (reached the superstep synchronisation barrier), we apply the transform parameters found for each view, thus bringing the entire set into better alignment with one another, completing our barrier synchronisation step. We then redistribute the tasks to perform a re-estimation of the sampled surface, using the new view-point positions, for each view in parallel. This typically results in a tighter, more accurate, estimation of the surface (see Chapter 3 for accuracy experiments). We iterate this process for S supersteps until viewpoint registration convergence has been reached. Convergence can be identified by looking at residual point alignment error metrics or the magnitude of the transforms being found by each task optimisation. In practice convergence is usually reached within $S = 10$ supersteps however for the purposes of the timing experiments in section 4.4 we use up to $S = 30$ supersteps.

4.5.1 *Experimental setup*

We evaluate this parallel alignment strategy quantitatively on synthetic and real range sensor data where we find that we have competitive registration accuracy with existing frameworks for this task. See Chapter 3 for registration accuracy results. Here we evaluate application speed up due to parallelisation. As discussed we are able to register all views simultaneously by taking advantage of many cluster nodes, and thus distribute the work. Here we explore various distributed *task* and *superstep* configurations and look at the performance gained by making use of a distributed system compared to performing the work on a single node. In the case of the single CPU experiments we register each scan serially using an individual cluster node and then find the related surface estimates once rigid transforms have been found for all scans.

We record runtime results as follows: for Single CPU results no job queueing is involved as the algorithm performs the registration of each scan in series until completion. The time reported is the total time required to register N viewpoints in series over S supersteps. For the parallel distributed experiments we measure the time taken in two ways. As discussed in section 4.3.1, the distributed system we make use of employs a multi-user job queueing system. Firstly, we measure the wall-clock time by recording the total real-world time required from the point of submitting our work to the job queue until the job is complete (when the registration of all viewpoints V_i has converged in this case). Here, job queueing (non-working) time cost may be incurred by each individual distributed task, (the alignment of a single view V_i to the related surface estimate to find the optimal pose transform T_{θ_i}). In Table 5 this timing result is referred to as "ECDF wall-clock time". The second distributed timing measure excludes this queueing (non-working) time and for each superstep finding the maximum task length of an individual distributed task (scan alignment) in a similar measurement process to that outlined in section 4.4. The time reported for this second metric is then the sum of the maximum task lengths over the total number of supersteps, we call this the "Distributed ideal time". We consider this to be an accurate assessment of the computation time required, as each superstep must wait for all member distributed tasks to finish before it may apply the global synchronisation step and then launch the following set of distributed tasks. This second metric excludes real-world queueing time. Furthermore, for this experiment, we have sufficient worker nodes to process all distributed tasks

in a superstep concurrently (true in the case of our current HPC cluster). These measurements allow us to compare the optimal theoretical performance gain to real-world speed up, achieved in practice on our multi-user system.

4.5.1.1 *Performance evaluation*

The success of employing an HPC system to solve computationally demanding problems resulting from large real-world data sets depends on the system architecture (*e.g.* number of available processors) and algorithmic design. The performance of an algorithm on an HPC system can be evaluated by calculating the speed-up provided over a single node or single CPU system. Here we use speed-up S_p and efficiency E_p (Equations 25 and 26) to show the improvement we achieve by formulating computer vision problems under our task farming framework. Assuming that the speed of processors and the network is constant; then speed-up [12, 81] is often defined as:

$$S_p = \frac{T_1}{T_p} \tag{25}$$

where p is the number of participating processors, T_1 is the computational time needed for sequential algorithm execution and T_p is the execution time required by the parallel algorithm when making use of p processors. Ideal (linear) speed-up is obtained in the case $S_p = p$. Although super linear speed-up is possible in some cases (*e.g.* due to cache effects in multi-core systems), when using task farming and an HPC cluster we consider linear speed-up as ideal scalability. In the linear speed-up case, doubling the number of processors p will double the speed-up S_p (halving the required execution time T_p). The second, related performance metric we make use of is efficiency (Equation 26). The E_p metric, typically in range $[0..1]$ attempts to estimate how well utilised p processors are when solving the problem at hand compared to how much time is spent on activities such as processor communication and synchronisation.

$$E_p = \frac{S_p}{p} = \frac{T_1}{pT_p} \tag{26}$$

For our viewpoint registration algorithm Table 5 shows that, in experiments performing only a single superstep (surface estimation), when we compare the serial and distributed computation times (excluding job queueing time) we are able to achieve significant speed up in each case (where here $p = 5, 20$ and T_1, T_5 and T_{20} timings are in

minutes) with $S_5 = \frac{37.26}{8.74} = 4.26$ and $S_{20} = \frac{95.38}{7.74} = 12.32$. We note that the experiment aligning fewer viewpoints, using fewer nodes ($|\{V_i\}| = 5$, $p = 5$, $S = 1$) achieves a result closer to optimal speed-up (and efficiency). We reason that a longer maximum task time (the superstep time) is likely to be observed for the larger experiment ($|\{V_i\}| = 20$, $p = 20$, $S = 1$) as it contains more distributed tasks per superstep. This point holds in practice here and was explored during our predictive model formulation and related scalability experiments in section 4.3.3. Table 5 also shows the same task set sizes ($|\{V_i\}| = 5, 20$) but with multiple supersteps ($S = 5$), which achieve slightly improved speed-up and efficiency performance: $S_5 = \frac{176.06}{39.12} = 4.50$ and $S_{20} = \frac{835.02}{52.40} = 15.94$. Again our hybrid model predictions come within 10% of the measured values in each case and we include ECDF wall-clock time results in the distributed experiments for completeness. The time required to align 20 range image viewpoints over 5 supersteps using our simultaneous method can be effectively reduced from $\sim$ 14 hours to fifty minutes.

Table 5: Multi-view registration algorithm timing results: single CPU versus distributed cluster.

	Single CPU (min)	Distributed ECDF wall-clock time (min)	Distributed ECDF ideal time (min)	Model prediction (min) (eq. 24)	S_p (eq. 25)
5 views 1 superstep	37.26	10.77	8.74	8.37	4.26
20 views 1 superstep	95.38	10.89	7.74	8.28	12.32
5 views 5 supersteps	176.06	49.22	39.12	36.06	4.50
20 views 5 supersteps	835.02	185.94	52.40	49.37	15.94

All implementation examples presented in this work make use of Matlab and we find that the prerequisites for writing parallel code under the Distributed Computing Toolbox (DCT) from MathWorks [174] are relatively low. There is no need for the developer to instruct cluster machines on how to communicate, which parts of the code to execute or how to assemble end results. We find that this provides a straightforward and intuitive approach to parallelising computationally demanding applications in a reasonable time frame. Parallelisation under this simple task farming framework results in potentially huge time savings without requiring extensive task or data parallelism knowledge.

In the following Chapter (Chapter 5) we explore registering 3D point cloud data captured using the Microsoft Kinect camera [183]. The Kinect is a structured light laser scanner that obtains a coloured 3D point cloud, with more than 300000 points

at a frame rate of $30Hz$ providing new standards in the quantities of rapidly available depth data. Consumer-grade, affordable sensors such as the Kinect are paving the way for a new era in computer vision that makes use of depth information modalities in ways previously impossible due to limitations on sensor speeds and costs. The potential advantages that fast, inexpensive yet accurate depth sensors can enable are considerable in many applications. However such sensors also bring associated challenges in the area of being able to successfully and gracefully handle the large volumes and sets of *(*e.g.) point cloud data generated by these sensors. This provides impetus for methods and techniques capable of processing large sets of point cloud data. In the following Chapter one promising route to satisfy these requirements is explored. By making use of the techniques introduced so far in this thesis we experimentally explore registering very large collections of point cloud data, captured from varying viewpoints using a Kinect sensor, and analyse potential applications.

4.6 DISCUSSION

In this Chapter, we have formulated a Semi-Synchronised Task Farming framework (SSTF) for solving computationally intensive problems where independent problem components can be distributed as parallel tasks to an HPC cluster. Following a round of task computation, results are collated and communicated. These results can then influence the initialisation and parameterisation of the following round of task distribution. This iterative procedure of task distribution and result collation leads to global problem solutions. The SSTF framework is complemented by a timing model used to predict overall application completion time for problems that are formulated using our task distribution strategy. We validate this model using simulation and experimental results and find it to be sufficiently accurate, providing a simple tool that can be utilised when estimating the time requirements of computationally expensive applications.

As might be expected, our experimental results illustrate that processing data sets using an algorithm formulated under our distributed framework, and deployed on an HPC cluster, obtain significant time saving over single node computation due to vast gains in speed-up. We note that, in practice, the human effort required to move from an original serial algorithm implementation to a distributed task farming approach is very reasonable. By making use of SGE to handle the task queueing system and

allowing developers to concentrate on domain specific problem aspects we find that we are able to convert a serial code implementation in a feasible time frame (*e.g.* one week). By employing parallel-friendly programming languages, master-slave communication is also hidden from the developer allowing them to again focus solely on domain specific problems.

Specifically the performance enhancement obtained when utilising SSTF to guide a parallel implementation of our (previously introduced, Chapter 3) point set registration algorithm is explored and documented. Throughput achieved, using our task farming framework, is compared with that of implementations using only a single compute node. In the application experimented with we find near linear speed-up improvement in the number of participating processors p over the related serial implementation. Also, in the case of the problem investigated, we are able to provide timing model cost predictions that are always within $\sim 10\%$ of the execution time required in practice. We therefore consider this timing model a useful predictive tool.

Distributed computing on HPC clusters offers an attractive option when compared to expensive integrated mainframe solutions. The main advantages of HPC clustering include distributed robustness and the ease of cluster scalability. When using an HPC cluster to accelerate the rate that we are able to solve computationally expensive problems, the factors of data set size and algorithm design play important roles in determining the degree of success in parallelising an application. Our framework allows the performance of a distributed algorithm implementation, on a given architecture, to be predictable. Using our SSTF framework and simple timing parameters obtained from the implementation under evaluation allow for reasoning about program design at an early stage.

Possible extensions and avenues for future work include implementing solutions using our SSTF framework in conjunction with faster compiled languages (*e.g.* C/C++) and applying such solutions to time critical applications. Additionally, extending our performance modelling treatment, to account for heterogeneous processors, would likely improve the model predictive accuracy and power. Related extensions might take the form of re-examining individual task time fitting using more sophisticated distributions to improve modelling in the heterogeneous processor case (*e.g.* employing distribution mixtures). Finally during the experimental work performed here it was noted that in

practice there is often contention between speed-up and efficiency. Future work could aim to find optimal trade-off generalisations from the specific cases presented here.

In summary the work in this Chapter introduces a straightforward parallelisation strategy that produces effective methods for solving computationally expensive problems offering vast wall-clock time savings over serial approaches. Our main contributions in this Chapter include the proposed strategy for formulating demanding problems that require a level of communication between subtasks and this strategy is explored experimentally using example problems from the computer vision domain that exhibit large time savings in practice when compared to serial implementations. Additionally, by taking inspiration from previous work regarding both analytical modelling (the Bulk Synchronous Parallel model [267]) and simulation based performance models we propose a timing performance prediction formula that we evaluate in simulation and practice. We show that this formula is able to accurately predict computational costs for distributed algorithm implementations thus providing a useful tool that can be utilised when planning distributed computational work. In the following Chapter (Chapter 5) we make use of the framework and tools introduced here by registering large volumes and sets of point cloud data, generated by consumer-grade depth sensors. By exploring the registration of very large collections of point cloud data in feasible time frames we are able to analyse the potential benefit of utilising the distributed strategies introduced in this Chapter.

Part V

LARGE SCALE POINT CLOUD REGISTRATION

5

LARGE SCALE POINT CLOUD REGISTRATION

5.1 INTRODUCTION

Chapter 3 presented a method for the simultaneous registration of multi-view range images using adaptive kernel density estimation. By producing a data-driven density estimate of object shape we provide a method that can be utilised to register point clouds simultaneously into a global coordinate frame. The performance of our KDE based technique on the multi-view registration task generally depends on both the initial coarse point cloud alignment provided and the extent to which the method is able to handle possibly noisy depth measurements whilst aligning overlapping views. In general, when we apply the technique to data sets containing viewpoints that afford large amounts of spatial overlap (with a reasonable initial coarse alignment) we are able to maintain and refine global object shape whilst converging on a tight and robust view alignment. By making use of all views at once to infer a model of global object shape, and allowing all views to improve their spatial positions simultaneously with respect to this model, the position of each view at a given time point is constrained and guided to a pose that is influenced by the current positions of all overlapping views. By making use of many overlapping views captured from the same physical portion of a surface or object we claim that through redundant sampling and measurement we are able to reinforce the correct view pose and improve registration performance in comparison with other techniques that *e.g.* directly locally minimise point pair distances.

Chapter 3 demonstrated that our multi-view registration framework produce registration results comparable to the state-of-the-art using data sets with relatively small view counts and highlighted several of the framework's desirable characteristics: in the case of having many overlapping views we are able to implicitly reinforce convergence

to the correct global object shape, improving registration accuracy. Additionally the method generalises to objects of any topology and does not require a training stage due to the non-parametric approach taken to density estimation, that will typically improve estimates and accuracy as more point samples are afforded.

In Chapter 4 we presented a framework that enables the distribution of computationally expensive problems that can be instantiated using non-independent (yet parallelisable) subtasks. We coupled this with a modelling process capable of predicting the available speed-up benefits available to an algorithm realised under this Semi-Synchronised Task Farming (SSTF) framework.

An intrinsic property of non-parametric density estimation dictates that estimation quality improves as the number of available samples increases. This fact provides the motivation for this chapter in which we explore whether we are able to improve view registration (and related model reconstruction) quality by applying our registration framework to large sets of object and scene viewpoints that potentially contain multiple and redundant depth samples of the corresponding physical points from varying views. In this regard we investigate model reconstruction quality as the number of available viewpoints increase. A second non-parametric model estimation property dictates that the cost of building models will increase as the number of available samples to be utilised increases. In this chapter we mitigate this foreseen computational cost increase by instantiating our registration framework under the proposed SSTF distributed computation model.

By combining the frameworks of the previous two chapters, we form a strong registration method that couples the previously noted advantages of registering viewpoints utilising a non-parametric model estimation technique and simultaneous view-pose alignment strategy with a framework capable of handling the simultaneous registration of view-sets containing large numbers of views. View sets experimented with are of an order of magnitude that is infeasibly large for traditional serial and sequential point cloud registration methods. In this chapter we explore potential available benefits when building models of objects and scenes from data sources containing viewpoint counts that are $1-2$ orders of magnitude greater than traditionally available. Large view collections are explored in this study and it is well understood that non-parametric density estimates tend to improve the accuracy of their estimates as the number of available samples increases. Considering these points, we hypothesise that:

1. *"Implementing KDE multi-view registration (chapter 3) under the SSTF framework (chapter 4), allows the scaling of view registration to successfully undertake problem instances consisting of view-sets* $1-2$ *orders of magnitude larger than traditionally considered. By enabling scalable multi-view registration strategies, that generalises well to differing sensor modalities, it becomes possible to successfully register high view-count datasets afforded by contemporary depth sensors targetting diverse physical objects."*

2. *"A registration method, utilising non-parametric surface inference and* soft *correspondence based strategies, is able to take advantage of information provided by redundant point sampling of object surfaces for the purpose of improving registration tolerance to sampling noise and coarse seed configuration."*

3. *"Increasing view-set order of magnitude, when undertaking point cloud registration, affords model reconstruction accuracy benefits over utilising sparse view-sets."*

Evidence in support of these hypotheses is collected and presented in this chapter. We illustrate how model reconstructions can be obtained by performing distributed view pose optimisation. The benefits of facilitating point set registration of large view-sets in a feasible distributed manner are explored. We apply our distributed registration model to several challenging large view-set data sets and provide evidence to support the claim that model reconstruction quality obtainable from large depth image view-sets improves over that of sparse view-sets. By applying our multi-view registration strategy to large view-sets obtained using sensors such as the Kinect, high speed stereo camera rigs and synthetic data sets we provide evidence of the potential quantitative benefits achievable when utilising large view-sets during the process of acquiring 3D object models within a modelling from range data pipeline. Using the framework introduced in chapter 4 (and therefore providing a multi-core implementation) we are able to keep wall-clock run times reasonable when working with high order of magnitude frame counts while maintaining high registration accuracy. Specifically we provide evidence in support of *hypothesis* 1 in sections 5.4.1, 5.4.3 and 5.4.4; *hypothesis* 2 in section 5.4.2.1 and 5.5 and *hypothesis* 3 in section 5.4.2.2. In summary this chapter presents a solution for the global registration of large collections (hundreds of viewpoints) of dense range images

(thousands to hundreds of thousands of depth sample points) as part of a modern, high-quality, 3D object modelling pipeline.

The remainder of this chapter is structured as follows: In section 5.2 we detail an approach for the preliminary task of automated coarse alignment when using large view-sets and section 5.3 introduces some additional point cloud registration considerations when the task involves high order of magnitude view sets. We describe our experimental results in section 5.4. Sections 5.4.1 and 5.4.2 provide evidence concerning the benefits of utilising large view-set data, such as that typically afforded by modern consumer grade depth sensors and the remaining subsection concerns comparing and contrasting the robustness and accuracy of our large viewpoint registration methodology with existing work in the literature. Finally section 5.5 concludes the chapter with some discussion.

5.2 AUTOMATED COARSE ALIGNMENT FOR LARGE VIEW-SETS

There are many capable sensing techniques for acquiring 3D data (*e.g.* laser scanners, tactile probes, structured light, stereo cameras, time-of-flight *etc*) and many contemporary sensors offer large depth data sets in terms of both dense sampling and high frame-rate view / image capture. Depth sensing mechanics are varied. However a common pipeline of operation for taking acquired depth data and producing a usable geometrical model is well established. The crucial step of depth image viewpoint registration is (as chapter 3 explains) usually split into the stages of finding an initial coarse global alignment followed by refining and optimising this alignment among viewpoints.

The task of computing initial alignments between samples, bringing all views into a single frame of reference, has historically been an active area of research. The general formulation of the coarse alignment problem, making no assumptions about scene object features or initial approximate registration is notoriously difficult to solve robustly. A broad history of automatic coarse alignment techniques include early work utilising the frequency domain [63], interest points [219, 247, 89], Harmonic maps [284], Spin Images [139, 131, 130], template set matching [113], computing principal axes [75] and exhaustive correspondence point search [51, 52]. More recent work [148, 50] also considers line-based and PCA based approaches (see [226] for a comprehensive review).

In sophisticated systems, coarse alignment can be aided by an ability to track sensor position and orientation and by affording approximate tracking (now with contempo-

rary, cheap hardware) in an attempt to alleviate an infeasibly large and unbounded pose space search. This has been performed using both physical coordinate measurement devices, tracking position and orientation and with optical tracking, deriving natural scene image features from (*e.g.* intensity data co-aligned to the depth data) or by manually augmenting the scene using physical fiducial markers.

Simpler depth capture systems often perform initial coarse frame of reference alignment by scanning objects on a turntable, providing a simple and cheap solution. This approach limits the size and complexity of scanable objects and, since the system produces cylindrical view sets, capture failure may occur where self-occluding objects are studied. Additionally this capture process may result in data sets with no data pertaining to the top or bottom of a target object (not viewed by the stationary sensor).

Finally a large number of pipelines rely on interactive manual alignment: a human is given control over identifying and selecting three (or more) matching feature points between views (thus allowing a closed-form rigid spatial transform to be derived) or allowing full control over view pose space parameters from which manual coarse alignment can be performed.

Whether using controlled motion, feature matching techniques or manual alignment, attaining the same degree of accuracy as sensor depth measurement is typically not achievable [20]. Initial alignment is therefore often refined by a following *fine-registration* stage. In order to explore the ability of the proposed *fine-registration* method (chapter 3) to register large view-sets, we augment our depth data to geometrical model pipeline with simple, automated *coarse-alignment* techniques capable of providing view-pose seed configurations as input for our simultaneous view registration framework.

In order to perform *fine-registration* with large sets of object views here we instantiate a simple full depth image to model pipeline by utilising common autonomous coarse alignment methods. Previously (*c.f.* Chapter 3) a manual coarse alignment was used for seeding viewpoint poses however, this task quickly becomes infeasible when exploring large problem instances where (1) a temporal view order is unknown or cannot be used to successfully infer a global frame of reference or (2) the number of viewpoints is of an order of magnitude that renders manual coarse alignment infeasible. By utilising well understood automated coarse alignment techniques in combination with depth data from an array of devices, this chapter instantiates a simple fully automated

pipeline capable of model reconstruction utilising depth data obtained from large sets of viewpoints.

Simple controlled motion is employed by pairing contemporary depth sensors ([73], [183]) with a turntable to provide cheap, fast large view-set depth data generation. The noted limitations of turntable capture are addressed by supplementing this process with a commonly used coarse alignment method involving point-to-point correspondence and Spin Images (originally proposed by Johnson [139]). In the following section we briefly provide detail of the implemented coarse alignment steps that offer coarse pose-seeding of large view-set depth data from a variety of range-finders.

Several noteworthy high quality full depth-data-to-model pipelines have been proposed previously. The real-time model acquisition system of Rusinkiewicz et al. [222, 223] affords interactive (real-time) model reconstruction with a structured-light range finder and more recently KinectFusion [135, 190] introduces similar rapid surface reconstruction functionality using the Kinect sensor. The main advantage that interactive frame-rates bring is the ability to offer a live visualised model preview during scanning that in turn facilitates valuable feedback relating to areas of a scene or object still to be scanned (useful for addressing remaining model holes *etc*). Several design decisions and concessions are made to afford these interactive frame-rates.

The earlier pipeline of Rusinkiewicz et al. make use of the natural 2D array organisation (pixel connectivity) of depth images and implement a projection-based ICP [221, 25] strategy. By projecting line-of-sight rays into range maps, matching point-pairs between frames are found simply by indexing into the 2D pixel array and avoiding the comparatively slow 3D closest point search. Additionally real-time model rendering is achieved by computationally frugal splatting [291] to give the appearance of merged surfaces without the need to triangulate points or reconstruct a consistent polygon mesh. User input in the form of manual alignment using anchor scans is relied on to correct misalignment errors and real-time speed is achieved. Reconstructions are good enough to guide users for object scanning feedback (*e.g.* hole filling) but intermediary models are admittedly not able to match the quality of offline state-of-the-art registration and reconstruction algorithms. To address reconstruction quality, a final *offline* globally-optimal registration component is offered at the conclusion of the scanning process using the technique of [210] to afford high quality final results[1].

1 This global registration technique is directly compared with the current work (see section 5.4.2.1)

By choosing not to perform fast *e.g.* line-of-sight projection point matching (using 2D depth images) we sacrifice real-time performance but in return become agnostic to range-finder source. By not requiring any point sample connectivity information our method generalises to register multi-view point cloud data obtained from any sensor modality (*i.e.* where 2D range images and neighbourhood connectivity information are not available). Run-time performance priorities also influence the KinectFusion work where highly parallel general purpose GPU (GPGPU) techniques are used to maintain a running scene model with a voxel-based signed distance function representation and a parallelised implementation of ICP [21] again neutralises costly nearest neighbour point search during pairwise view registration. Maintaining acceptable real-time frame rates for this task involves accumulating large amounts of depth data such that rapid merging or discarding of redundant data is required. Conversely in this work, at the cost of real-time operability, we retain large amounts of data for offline processing and explore the benefits that not explicitly discarding redundant sampling information is able to afford (*c.f.* experimental section 5.4.2.3).

In conclusion fast, real-time full modelling pipelines exist however this work concerns high quality simultaneous multi-view registration, the expensive process that can be considered useful for refining or finalising results offered by instances of the outlined multi-step process. A basic implementation of the discussed pipeline is instantiated in this work to facilitate experiments. This involves simple common coarse alignment methodology, allowing a focus on improving the global simultaneous *fine-registration* step in terms of quality and feasibility. Additionally, surface reconstruction techniques are used in section 5.4.4 to help visually assess the obtained multi-view registration results. There are many algorithms available that produce high quality surfaces. Due in part to the previously discussed ability of the proposed registration strategy to implicitly handle outliers (*c.f.* sections 3.3.4, 5.4.2.3) we choose not to implement any specific point outlier removal or integration process pre-surface reconstruction. This influences the selection of Poisson surface reconstruction [145] to complete our basic pipeline, a popular implicit surfacing method, capable of creating smooth surfaces and robustly approximating noisy data.

5.2.1 *Coarse alignment using local descriptors*

Spin Images [139] are a popular local 3D shape descriptor that employ 2D histograms to evaluate the spatial neighbourhood of selected interest points in a point cloud (or on a surface). A Spin Image provides a direction and orientation invariant signature associated with each selected location. The statistics of each descriptor are influenced by both the size of the spatial neighbourhood considered and the granularity of the 2D histogram made use of. These variables can be adjusted to obtain the desirable local descriptor properties of being unique and distinguishable, yet repeatable, point signatures. Interested readers should review [139] for further detail regarding Spin Image construction. The spatial relationships between these local descriptors are stored in the geometry of a given point cloud. By finding potential matching descriptors between point clouds (using *e.g.* cross-correlation) and producing a set of sparse point correspondences, an initial coarse alignment can be found by locating a common subset of matching points. Point matches can be used to estimate rigid spatial transforms, in closed form[2]. We emphasise that at this stage we do not seek perfect correspondences (or therefore a transform that results in a perfect alignment), just a sparse set of reasonable matches to determine a transform that provides a coarse seed alignment between point clouds.

Figure 47 shows Spin Image [139] point descriptors calculated at selected points (using a single point cloud from the *Pipe* data set [214]). This feature involves creating 2D histograms. Calculating distinct local features at varying spatial locations provides evidence of the descriptor ability to provide unique (yet repeatable) local features. Adjusting 2D histogram bin counts (effectively the descriptor resolution, 15×15 bins in this instance) and the local spatial region considered influence desirable properties of feature uniqueness and repeatability rate.

2 Several popular closed form solutions provide a rigid transform determined using a set of corresponding points. Solutions differ in their transformation representation and method of criterion function minimisation. See [162] for a detailed review.

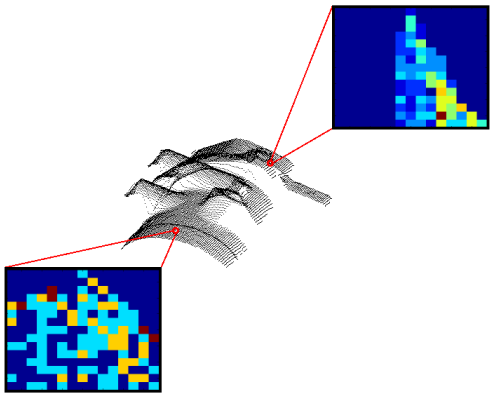


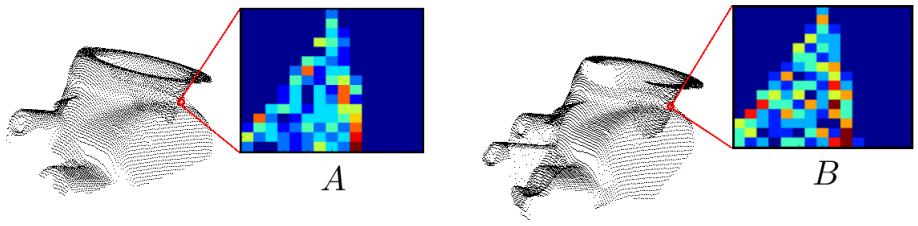
Figure 47: Spin Image [139] local descriptors calculated from a (single view) point cloud using the *Pipe* data set [214]. The diversity of descriptors gathered from geometrically distinct locations confirm the ability to provide potentially unique (yet repeatable) local features.

In Figure 48a we exhibit a Spin Image defined at a sample point, chosen manually for illustrative purpose, and an approximately equal point on a (distinct) view of the same object (48b). The point correspondence does not likely provide a perfect match regarding location on the physical object (due to sensor quantization, measurement noise, manual error *etc.*) however the 2D histograms provide visually similar results and the similarity of the descriptors can be computed by a standard image cross correlation metric (Eq 27) between images A, B.

$$Corr(A,B) = \frac{\Sigma_i\Sigma_j(A_{i,j}-\mu_A)(B_{i,j}-\mu_B)}{\sqrt{(\Sigma_i\Sigma_j(A_{i,j}-\mu_A)^2)(\Sigma_i\Sigma_j(B_{i,j}-\mu_B)^2)}} \tag{27}$$

Using this simple comparison metric (Eq 27), perfectly correlated (identical) feature "images" produce $Corr(A,B) = 1.0$ and highly correlated point matches indicate a good chance of a valid location match between views ($Corr(A,B) \approx 0.91$ in the Figure 48 example). Using this simple correlation based similarity metric and a standard RANSAC [91] step to find the best consistent correspondence model provides a sparse strategy for finding initial correspondence (and alignment) between point clouds. This strategy can be performed in a standard chain-pairwise fashion between views when satisfactory feature point correspondences exist to produce a simple autonomous coarse alignment strategy. It was found that this simple strategy works well when considering view points of objects exhibiting unique, distinctive and varied geometrical features. Objects that contain *e.g.* many symmetrical parts are more likely to produce poor

coarse alignment results. Where necessary, this feature based coarse alignment strategy is augmented with the additional approach described in the following section (5.2.2). Finally, manual hand alignment can be used to correct any remaining coarse alignment errors. Example coarse alignment seed results for point clouds, collected from various sensors, are found in section 5.4.



(a) (b)

Figure 48: Matching descriptors representing an identical physical object location between (distinct) point clouds. The point correspondence may not provide a perfect match regarding location on the physical object (due to sensor quantization, measurement noise, manual error) however the 2D histograms provide visually similar results and the similarity of the location is affirmed by a standard image cross correlation value of ~ 0.91 (Eq 27) between A, B. Feature point locations are chosen manually for illustrative purposes.

5.2.2 *Heuristic sequential coarse alignment*

The second simple coarse alignment seeding method we implement to precede *fine-registration* is applicable to large view-sets where a temporal view ordering is known and the trajectory of the sensor (or scene target) is also approximately known in advance or can be estimated. This technique is successfully applied when a view-set consists of many frames, with small temporal gaps, and the sensor (or target) follows a relatively simple or easily predictable path through the scene. An example scenario, utilised in this work, involves rigid object capture from many points of view using a fixed position depth sensor and the aid of a physical turntable device. By considering sensor frame capture rate and total capture duration (and introducing reasonable simplifying assumptions *e.g.* a constant turntable velocity) we can estimate the inter-frame rotational transforms exhibited over the duration of a complete object revolution. We use the estimated

rotational transforms between view frames to again perform a simple chain transform application between subsequent frames thus bringing all frames into a reasonable coarse alignment. This second coarse alignment seeding technique is applied to the data set made use of in section 5.4.1.1 and aids coarse alignment performance when tackling the noted difficulty of seeding viewpoint positions that contain many similar or symmetrical parts (see Figure 51 for coarse alignment examples). In the following section we briefly outline some considerations that occur when attempting to perform fine registration on the coarsely aligned large view-sets that result from the work described here (sections 5.2.1 and 5.2.2).

5.3 FINE REGISTRATION FOR LARGE VIEW-SETS

The process of acquiring high quality 3D models from large view-sets of range data typically require that a final global optimisation is performed in order to reduce and evenly distribute residual alignment error due to *e.g.* sensor noise or poor coarse initialisation and error propagation between consecutive views. A global registration step is often motivated by the resulting improved registration quality and successful solving of "loop closure" like problems. Desirable properties of a global registration stage include robustness to varied initial alignment configurations and computational feasibility. Computation time often becomes an issue in both the case of dense spatial sampling (high order of magnitude of points per point set) and the case of large collections of depth maps or point clouds (many point sets). Large point sets of both varieties are increasingly generated in specialised and professional application fields (*e.g.* biomedicine, orthopaedics, orthodontia, cultural heritage, reverse engineering and industrial design). In sections 5.3.1 and 5.3.2 we briefly outline the main considerations taken into account when undertaking high order of magnitude depth data registration. After consideration of large-view-set specific correspondence and optimisation concerns we proceed to explore the benefits of performing point set registration in the discussed large-scale problem instances.

5.3.1 *Point correspondences in large view-sets*

When point correspondences are known, estimating a transformation can be accomplished quickly (typically $\mathcal{O}(n)$ time). However, the step of finding correspondences often involves costly search with a naive (*i.e.* brute force) approach requiring $\mathcal{O}(n^2)$ time for n closest point correspondences. Chapters 2 and 3 (*e.g.* section 3.3.4) noted that k-d trees provide a popular data structure for storing point sets and can reduce the closest point correspondence search to logarithmic time. Even if geometric data structures are employed, computational expense often becomes challenging as n grows large (due to increasing view count or point sample resolution). The problem of point correspondence search for large view-set registration has been addressed from a number of directions (some review of common options were reviewed in Chapter 2, section 2.3.2).

In this chapter, we investigate the advantages that a *soft* correspondence strategy affords over a classical *hard* correspondence strategy when applied to large multi-view-set problem instances. We apply our registration strategy to problem instances where the aim is to mitigate the dual computational concerns that arise from (1) typically expensive *soft* correspondence strategies and (2) greatly increasing the number of (*hard* or *soft*) correspondences (n) required as the view-set size increases to an order of magnitude greater than that typically undertaken. We significantly reduce anticipated algorithm wall-clock runtime by implementing our simultaneous registration strategy (chapter 3) under our SSTF framework (chapter 4).

As detailed in chapter 3, our registration strategy optimises scan alignment by evaluating point positions in relation to a surface approximation. This approximation is inferred by applying non-parametric density estimation to the remaining scan views. By evaluating each point in a moving scan against this inferred continuous surface our registration quality measure benefits from the advantages that a *soft* point correspondence strategy offers (*e.g.* convergence from a wide basin of coarse seed alignment) and the ability to tackle problems that *hard* correspondences find challenging; *e.g. true* one-to-one point correspondences between scans may not exist due to (*e.g*) partial overlap, occlusion, sensor quantisation or sampling noise. Our density estimate provides a continuous, smooth and meaningful measure to evaluate points found at any spatial position in relation to our approximated surface.

The approach does however incur additional computational expense, often associated with a *soft* correspondence strategy. In the current work the evaluation of each point position requires information from a number of contributing points belonging to the inferred surface (the number of contributing points is dictated by the kernel bandwidth parameter discussed in chapter 3, section 3.5.2). Although finding nearest neighbouring points to define *hard* point pair correspondences is not required, defining a surface estimate does employ radius based search (or nearest neighbour search) when evaluating kernel contributions for density estimates. Additionally, as is often typical for non-parametric estimation, the cost of inferring a density increases with the number of available point samples. As detailed in chapter 3, our approach iteratively evaluates point positions belonging to a moving scan as revised transforms are applied and surface approximations are updated as each viewpoint independently finds an improved alignment with the current related surface approximation.

In summary, applying the *soft* point correspondences utilised in this work to large view-set problem instances is kept feasible through the use of the previously introduced SSTF framework (chapter 4). By combining our SSTF framework with the multi-view simultaneous registration strategy (chapter 3) we are able to (1) iteratively update our registration metric using *soft* correspondences, (2) consider the registration of all viewpoints in a global manner *simultaneously*, and (3) work with very large view-sets by distributing the computational cost.

5.3.2 *Transform space optimisation for large view-sets*

5.3.2.1 *High dimensionality global optimisation*

Global registration techniques often formulate and consider high dimensional optimisation problems in order to find optimal parameters in the joint transform space for all considered viewpoints simultaneously. The difficulty of solving such problems when the number of views become large is due in part to the increasingly high dimensionality of the search space. Global optimisations of this form typically scale the dimensionality of the search space linearly in the number of viewpoints N, regardless of the transform space representation employed. Searching high dimensional transform spaces to *globally* find optimal sets of (*e.g*) rotation matrices $\mathcal{R} \coloneqq [R_1, R_2, \ldots, R_N] \in \mathbb{R}^{3\times 3\times N}$

and translations $\mathcal{T} := [t_1, t_2, \ldots, t_N] \in \mathbb{R}^{3\times N}$ may become computationally infeasible. Recent work attempting to solve optimisation problems formulated in this manner, for large sets of views N, has utilised *e.g.* gradient information [27] to direct the search with respect to a registration quality measure. The expensive operation during such an optimisation is often the computation of an optimal search space descent direction and step size. The reason for this can be understood if one considers the dimensionality of Hessian matrices that must be derived from $\mathcal{R}$ and $\mathcal{T}$ for large N.

One route to address this problem considers avoiding the expensive high dimensional partial derivative computation. When many viewpoints are considered, [27] introduce a novel method to avoid full Hessian matrix calculation during each optimisation step using a decompositional approach. By defining the full Hessian H as the sum of a positive-semi definite term and a high dimensional term (that grows with the number of views considered) as originally described in [154] and then ignoring the calculation of the expensive latter term, it is reported that the former term alone can be utilised to estimate a trustworthy descent direction. Without this alteration global optimisation methods quickly become infeasible for data sets involving many viewpoints. Some experimental evidence supporting this point is given by the authors; even for relatively small view-sets ($N = 23$ views), it was observed that Hessian computation takes approximately half of the time required by a single iteration. The related full optimisation (performing full Hessian calculation [154]), was unable to complete view registration when tasked with aligning $N \geq 45$ viewpoints (see [27], pp. 448 for details).

5.3.2.2 *Simultaneous local optimisation*

The optimisation approach explored in this work optimises each viewpoint individually (yet simultaneously) in relation to independently inferred surface approximations. As detailed in chapter 3, sets of low dimensional transform space optimisations (per viewpoint) are performed. After optimising the position of each viewpoint (in a local $6D$ rigid transform space), the related surface approximations are then iteratively updated. These low dimensional optimisations are computationally feasible and typically solvable quickly in comparison to high dimensional alternatives. By taking the current pose of all other viewpoints into account (via surface inference) and alternating between this inference and transform space optimisations, the nature of the introduced procedure allows for the main benefits of full *simultaneous* registration to be retained; all views

are capable of adjusting their pose simultaneously and the pose of each view is implicitly constrained by the current poses of all locally overlapping views through surface inference. Alternating between updating surface approximations and allowing all viewpoints to move simultaneously in the transform space avoids the error propagation and accumulation problems commonly found in early sequential registration work whilst maintaining required feasibility when applying the technique to large view-sets.

By implementing the registration framework under our SSTF framework (chapter 4) we again map viewpoint pose optimisations onto compute cores individually and synchronise each superstep such that surface estimation is only performed after each simultaneous viewpoint pose local optimisation has completed (see section 4.5 for further detail). This strategy allows for viewset size scalability with available processing cores. While computational expense scales linearly with the number of views considered, in practice this allows for viewsets of sizes not typically considered to be utilised. Termination criteria for the registration is typically achieved by specifying a maximum number of supersteps, or observing an error metric until convergence (experiments in sections 5.4.1 and 5.4.2 respectively). Additionally it is possible to make termination decisions based upon the order of magnitude of the spatial transforms found during local pose search optimisation.

By not optimising all transform variables in a global space we can not guarantee global optimality (or theoretical convergence) and, therefore, sub-optimal final view pose configurations (local minima) are possible. Experimentally (see section 5.4) we find that, on the condition of reasonable coarse alignment seeding, this issue is not problematic in practice for the data sets explored in this work. By solving multiple local optimisation problems in low-dimensional spaces, pertaining to the pose of each viewpoint, visually satisfying and quantitatively competitive solutions are obtained (see section 5.4).

In summary, we are able to maintain the advantages of global registration by allowing all viewpoints to alter their pose simultaneously, (and thus react to pose alterations of other viewpoints). Yet by only performing local view-wise optimisations we reap the benefits of affordable (and potentially parallelisable) optimisation in low dimensional spaces. By parallelising these low dimensional transform space optimisations and iteratively improving surface estimates we are able to perform quasi-global simultaneous registration over large sets of viewpoints whilst keeping run times feasible for practical

applications. In the following section we present implementation details and potential benefits of large view-set point cloud registration.

5.4 LARGE VIEW-SET REGISTRATION EXPERIMENTS

Our registration method is evaluated using *large view-set* synthetic and real point cloud data and view alignment results are compared to the global registration technique proposed by Pulli [210], the previously utilised Procrustes method [262] and the coarse strategy outlined in section 5.2. We evaluate the experimental results using distance based registration quality metrics, model fitting and visual inspection.
We select these methods to compare against for the following reasons:

- The coarse alignment technique outlined in section 5.2 provides a simple baseline registration offering cheap initial alignment. By comparing fine registration results to this technique it becomes possible to evaluate how the examined methods are able to improve upon this initial alignment and explore the simple cost-benefit relationship of implementing a fine registration stage.
- The global registration technique proposed in [210] has proved popular for large view-set multi-view registration as evidenced by the fact that it has been adopted by the computer vision community and implemented in various pieces of end-user software such as Scanalyze [227]. The method can now be considered a classical benchmark for the task of global multi-view registration for the task of aligning large view-sets containing multiple overlapping range images since its introduction in [210].
- We again compare the Procrustes method [262] used in chapter 3 but due to the nature of the many view-sets considered in this chapter (and the available serial Matlab implementation of this algorithm) some data set down-sampling concessions are made (see 5.4.4.1 for details).

We consider a heterogeneous collection of 6 *large view-set* data where each set consists either directly of point clouds or a set of depth images (that are subsequently reprojected to point clouds using a standard pinhole camera model). Each data set (1) contains object samples from varying sensor viewpoints, (2) consist of a large number

Table 6: Statistics of large view-set point cloud data utilised for global simultaneous registration experiments.

Data set	Number of viewpoints	Mean # points per view	Data source	Depth sensor	KDE registration kernel bandwidth k-neighbourhood
Physical Tridecahedron	512	5000	Local capture	Kinect [183]	$k = 2560$
Synthetic Tridecahedron	250	5000	Local generation	Synthetic	$k = 1250$
42_fighter	258	19846	Stuttgart DB [214]	Synthetic	$k = 5120$
17_porsche	258	18547	Stuttgart DB	Synthetic	$k = 4150$
04_copter	258	7953	Stuttgart DB	Synthetic	$k = 2050$
Head bust	220	2209	Local capture	$24Hz$ stereo video [73]	$k = 485$

of viewpoints and (3) are collectively representative of a wide variety of real-world objects, sampled from a range of depth sensors. The data sets used in this chapter are briefly outlined in the following section. Objects captured are representative of real and challenging acquisition scenarios and present various surface and geometric properties. View sets are comprised of between 220 and 512 viewpoints and size dimensions of acquired physical objects range from $30cm$ to $\sim 80cm$. Data set statistics are summarised in Table 6 and in the following section we outline local object capture and depth sensor utilisation.

5.4.1 *Structured light sensors*

The Kinect is a consumer-grade structured light scanner capable of acquiring a RGB intensity image and (temporally interleaved) depth map from which a coloured 3D point cloud can be derived (see Figure 49 for an example RGB-D image frame). Kinect point clouds contain ~ 300000 point samples and can be captured at a frame rate of $\sim 25Hz$. The optimal range between sensor and target is typically ~ 1.2 to $3.5m$ [149]. The computer vision community has found that the Kinect enables depth sensing applications that extend far beyond the gaming functionality that the sensor was initially introduced for [117, 285]. Fast depth sensing can now be performed at low cost and sensors are priced competitively when compared to many traditional depth capture devices such as stereo or time-of-flight (TOF) cameras (*e.g.* [73], [243]). The Kinect camera is therefore well suited for tasks such as robotic navigation in workplace or domestic environments [250] and 3D object measurement and capture [190].

5.4.1.1 *Structured light sensors: data capture*

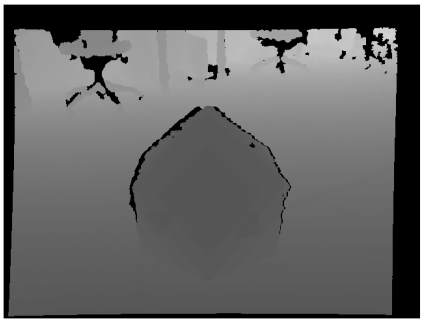
Initial local object capture is performed using the Kinect sensor [183]. A local physical tridecahedron object, a 13 sided polyhedron, with edge length = $\sim$ 30cm (see Figure 49a) was augmented with additional structure by attaching small spheres (table tennis balls) in various configurations in order to reduce planar object symmetries and increase shape complexity. The object is placed on a turntable and is captured using a stationary Kinect (while the object rotates), thus providing an example of a simple, largely convex, physical shape from which multi-view point clouds were obtained. Open source software [35] is used to retrieve RGB images and corresponding depth maps of objects placed on the turntable and rotated such that a target is captured from multiple points of view covering each object side.

Previous work has shown that additional information channels *e.g.* intensity information in conjunction with geometric information can be used to improve the accuracy of (pair-wise) view registration *e.g.* [259]. However the physical objects used in our Kinect experiments were simple in shape (semi-regular polyhedra) and contained uniform face colour therefore registration using only geometric characteristics could be employed successfully. Utilising simple convex objects allows for shape ground truth to be easily obtained *e.g.* physical measurement of object side lengths and angles that can be used to provide quantitative evaluation of the quality of the multi-view registration process. The open source capture software [35] is used to capture $\sim$ 20 seconds of footage during which the sensor position is kept fixed and the turntable (on which the object resides) is rotated such that multiple object views are captured containing each object face. This results in $\sim$ 500 depth images that are converted to point clouds using a standard pinhole camera model. This affords a simple, cheap and fast method for capturing local physical objects and potentially creating accurate 3D models. The experimental object is relatively textureless and uniform in colour, thus providing an example where a structured light depth sensor combined with registration techniques, making use of geometrical information alone, prove appropriate.

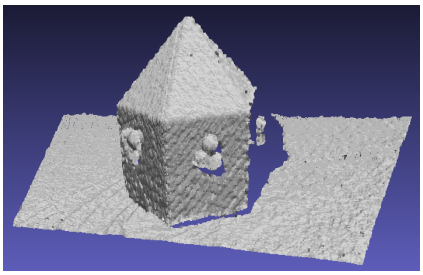
Example RGB and depth image frames and the resulting reprojected point cloud from a single view point are found in Figure 49. Pre-processing of the point cloud data (depth images reprojected to 3D space) involves segmenting out parts of the scene that do not belong to the target object. Here, manual segmentation is utilised for removing ground planes (floor) and background, see Figure 49d for a representative result. Performing

this point cloud segmentation step algorithmically would provide a useful additional component to further automate the model reconstruction pipeline. After pre-processing is complete, the mean point set size per viewpoint for this data set is 23170 points (pre view sub-sampling). Finally, we axis align each Kinect point cloud frame such that the object is orthogonal to the $x - y$ plane, helping to compensate for the fact that data capture was performed without front-to-parallel sensor-object capture.

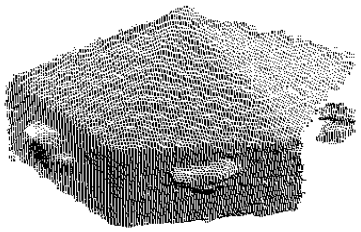
(a) Kinect RGB intensity image data. Tridecahedron object (frame 84 of 512).



(b) Aligned Kinect depth image data. Tridecahedron object (frame 84 of 512).



(c) Point cloud obtained by reprojecting a Kinect depth image to 3D space and masking out background depth data. Locally fitted surface normals are estimated and scene lighting applied for visualisation purposes.



(d) Post manual segmentation. Depth image data pertaining to the ground plane and background has been manually masked out. Note missing object surface data due to self-occlusion and sensor / object capture angle.

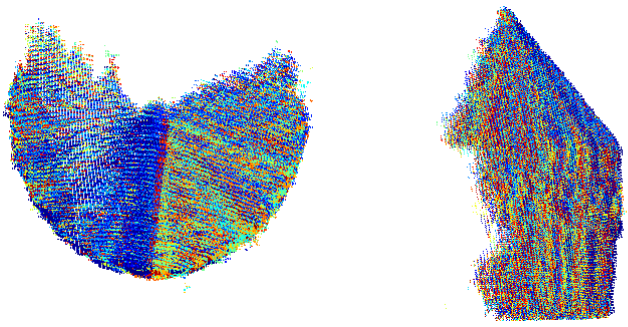
Figure 49: Object augmented with local surface structure (table tennis balls) to reduce planar symmetries and increase test shape complexity. Object is rotated on turntable allowing a stationary Kinect sensor to capture multiple views associated to each object face.

Capture duration and turntable rotation speed combine to afford ~ 5 complete object rotations in the particular experimental setup. This allows each physical side of

the object to be captured multiple times on multiple passes. Due in part to the nature of the capture process (relatively high turntable rotation speed, typical consumer grade structured-light frame capture rate), there is sensor noise and measurement error present in the depth data such that reprojected point clouds may contain measurement noise. It is conjectured that a probable contributing cause is turntable rotation introducing shape distortion as the object moves during the 1/25 second capture time.

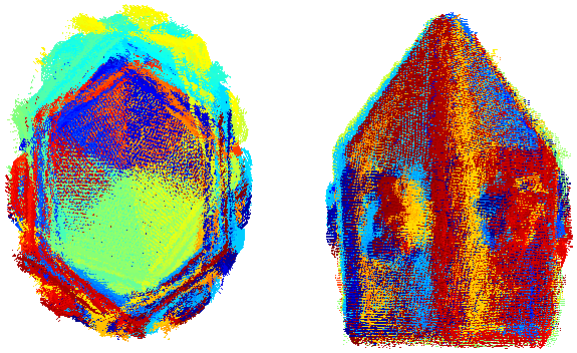
5.4.1.2 *Structured light sensors: multi-view registration*

Using the data capture method outlined above, objects can easily be coarsely aligned into a common frame of reference. Before coarse alignment is performed on the segmented and axis aligned point clouds, representing different views of the rotated tridecahedron, the views occupy overlapping 3D spatial location in world coordinates due to the turntable capture strategy. Figure 50 shows (a) normal to ground plane view where well defined object planar side panels are not easily distinguished due to the lack of a consistent reference frame and (b) orthogonal to ground plane normal. The point clouds overlap in world space, pre-coarse alignment. Point clouds are initially axis aligned using a fitted ground plane to account for the Kinect sensor capture angle. Initial coarse alignment is performed using the method outlined in section 5.2.2 providing a reasonable, yet inexact, coarse seed positioning in world frame coordinates (Figure 51). It can be observed that this coarse alignment strategy provides a reasonable, yet inexact, coarse seed positioning for all viewpoints in a world frame. This configuration provides our input for the fine registration algorithm experiments.



(a) (b)

Figure 50: Reprojected point clouds of the tridecahedron (512 viewpoints). One point cloud colour per viewpoint. Viewpoints are not coarsely aligned (or registered) in a coherent world frame, each view is in a local coordinate system as seen by the sensor. See text for detail.



(a) (b)

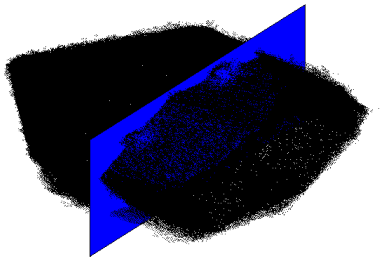
Figure 51: Coarse alignment applied to all 512 viewpoints of the tridecahedron object. The hexagonal polyhedron edges of the object shape begin to emerge and the vertical faces of the object are now visible in the reference frame. This view-set configuration is used as input for the registration algorithms.

Fine registration is performed on the tridecahedron view-set using the strategy described in chapter 3 and the methods ([210], [262]) outlined in section 5.4. Due to the size of the view-sets considered in this chapter the experimental work, considering the introduced registration strategy, makes use of locally available distributed compute resources. Each local view optimisation (as discussed in section 5.3.2.2) is distributed to an ECDF [82] compute core, thus framing the problem under the SSTF strategy introduced in chapter 4. When undertaking view registration for large view-set prob-

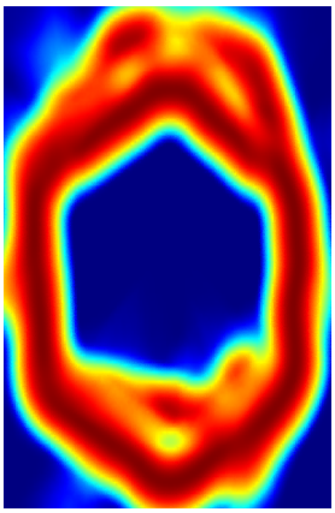
lem instances, we find that distributing the computational load and parallelising the work provides an effective solution that substantially reduces wall-clock time. Visual registration results for the Kinect data set are found in Figures 52, 53 and quantitative registration results (for all experimental data set) are summarised at the end of the chapter in section 5.5.

In Figure 52 density is represented by colours increasing from deep blue to red. The density estimate shows that the coarsely aligned configuration contains view misalignment and sensor noise. This is mitigated by using our adaptive kernel estimation process capable of smoothly estimating predominantly unimodal underlying surface structure that in turn helps to avoid mis-registration and view clique formation. The quality of alignment can be seen to visually improve after 10 simultaneous registration cycles. It can be seen from the density estimate that the true six sided shape of the polyhedron clearly takes form as registration improves. In this fashion planar density slices afford a further simple visual assessment of registration quality, providing an informative technique in cases where dense sampling may inhibit raw visual view-pose appraisal.

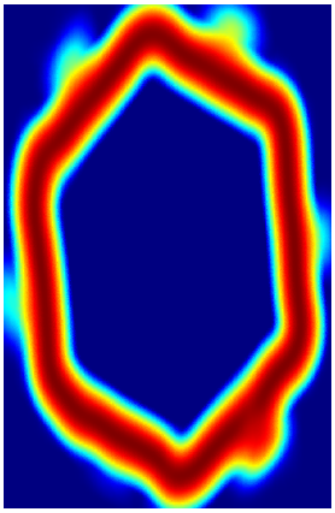
KDE registration is performed using algorithm parameters consistent with chapter 3, section 3.5; error metrics are found to converge within 10 superstep cycles (where each superstep involves the distributed optimisation of all view poses followed by the kernel density estimation process). The kernel size parameter (k-neighbourhood influencing kernel bandwidth) is chosen for large view-set scan collections using the method described in chapter 3, section 3.5.2 (however here, to accommodate view-sets that are typically an order of magnitude larger than those previously considered, we find decreasing k to $c.$ 0.1% of total point set magnitude suitable).



(a) Amalgamated tridecahedron point cloud view sets in the coarsely aligned configuration. Planar slice indicates the location that density estimation is queried at for visualisation.



(b) Density estimation for coarsely aligned configuration.

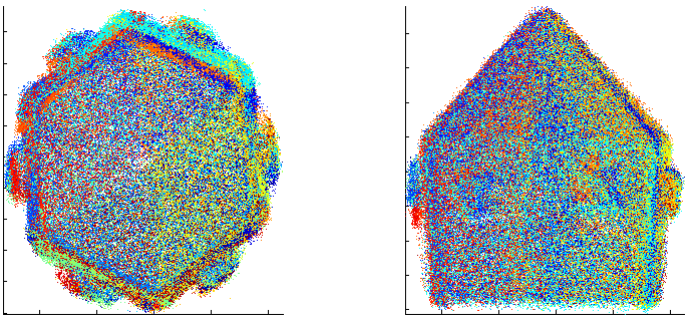


(c) Density estimation after 10 registration supersteps.

Figure 52: A planar slice through the tridecahedron amalgamated view set indicating density estimation location. See text for details.

Visual registration results for the tridecahedron view set are provided in Figure 53. It can be observed that for data sets containing hundreds of views, possessing sensor noise and relatively low sample resolution, a reasonable registration is found. The hexagonal polyhedron shape of the test object can be seen to emerge from the coarsely aligned view set (see Figure 53a). Greater perceived colour interpenetration typically denotes a better alignment result since each range image point set has a different colour. In principle, the lower the residual alignment error is, the better the colour interpenetration appears.

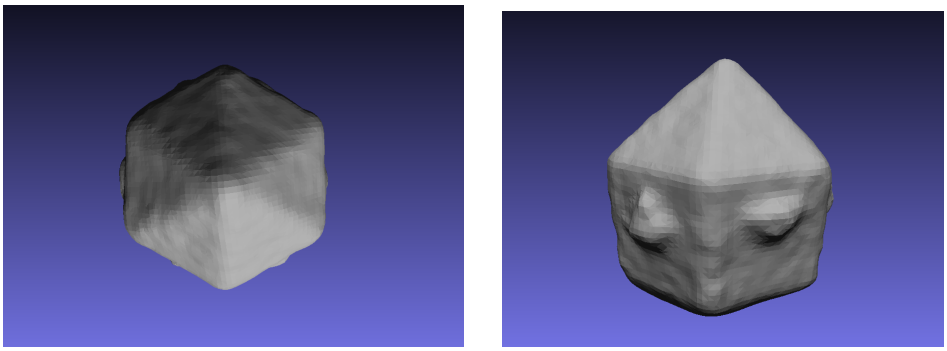
Inspecting the coarse registration in Figure 51b it is easy to identify a small number of distinct colours in the vertical planes of the object surface, even when 512 different viewpoints are present. Conversely Figure 53 exhibits that a greater interpenetration is obtained post multi-view registration.



(a) (b)

Figure 53: Final registration results achieved after applying our multi-view registration algorithm to the 512 viewpoints captured using the Kinect. The registration technique converges to a consistent object view configuration as demonstrated by perceived increased colour interpenetration over the coarsely aligned configuration. Viewing angles and camera vector directions as Figure 50. (Best viewed in colour.)

For illustrative purposes, a simple Poisson surface reconstruction [145] is applied to the full amalgamated, registered point set (Figure 54). It can be observed that, although additive, individual viewpoint sensor noise (and some minor misregistration) is present in the amalgamated point set, the resulting surface reconstruction produces a model that is visually recognisable as the tridecahedron object (*c.f.* object true RGB intensity image, Figure 49a). Small tri-sphere features lack some definition due to the simple view-amalgamation strategy and sensor resolution limitations. However global object shape is accurate in terms of side lengths, angle ratios and planar structure. Employing more advanced point set integration strategies (*c.f.* amalgamating all points) would likely aid reconstruction quality further.



(a) (b)

Figure 54: Poisson surface reconstruction [145] applied to the amalgamated, registered point set. Minor viewpoint sensor noise and misregistration still visible, however the resulting surface reconstruction produces a model that is visually recognisable as the tridecahedron object (*c.f.* Figure 49a). Small tri-sphere features lack definition due to the simple view-amalgamation strategy and sensor resolution limitations however global object shape is accurate in terms of side lengths, angle ratios and planar structure.

Further to visual registration assessment, Figure 55 shows the progress of the mean inter-point distance error metric defined previously (chapter 3, section 3.5.1.2) over 10 superstep iterations. A ground truth registration is not available, so simplifying assumptions again allow a heuristic approximation of the optimal mean inter-point distance error. By considering the number of point samples in the amalgamated point set (N) and evaluating the MeanDist $(2, N, 1)$ function (equation 18, section 3.5.1.2) we approximate the mean distance between an arbitrary reference point sample and its nearest neighbour under an optimal registration (assuming uniform point sampling density). Approximating the studied tridecahedron object surface by summing a collection of simple polygon areas (6 rectangular planes of area $600mm^2$ and 6 isosceles triangles of area $375mm^2$) a crude visible surface approximation of $5850mm^2$ is provided. Scaling the unit area of the MeanDist $(2, N, 1)$ result by this area approximation results in a sensible lower bound on registration accuracy (Figure 55, green dashed line). Performing multi-view registration on the tridecahedron data set achieves a mean inter-point error of within $\sim 11\%$ of this approximate lower bound. It is surmised that the remaining discrepancy is likely due to a combination of (1) crude surface area approximation (2) the uniform sampling density assumption (3) minor misalignment is still evident in the converged registration configuration (which may be partially caused by sample distortions in the Kinect data).

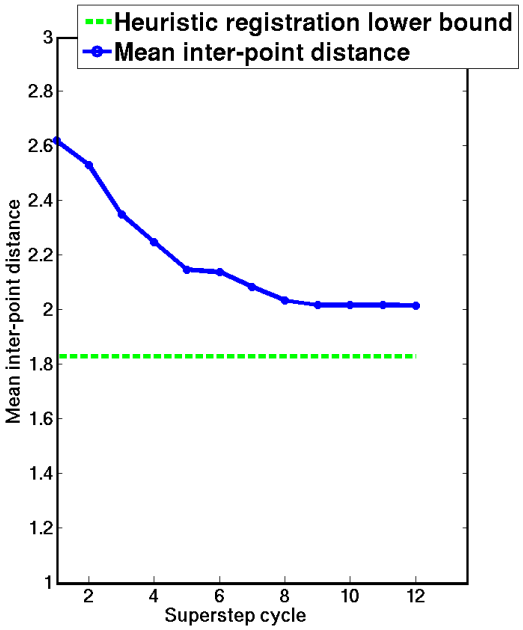


Figure 55: Kinect view-set: Mean inter-point distance error (defined in section 3.5.1.1) during iterative registration from coarse alignment seed. Optimal obtainable mean inter-point distance is defined using simple object surface area approximation and the assumption of uniform point sampling density (see text for further details).

5.4.1.3 *Structured light sensors: summary*

This initial experiment provides evidence in support of the claim regarding the ability of the introduced registration framework to scale to large viewsets afforded by contemporary depth sensors. Additionally, registration quality and timing comparison among the explored registration techniques, utilising this data set, are found in summary Tables 10 and 11 (at the end of the chapter). In the following section we explore a synthetic version of this data set where we have access to ground truth alignment and further evaluate registration quality with hundreds of views quantitatively.

5.4.2 *Synthetic data: data sets*

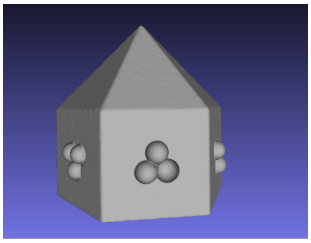
In addition to data sets captured using range sensors, we generated further synthetic 3D data sets with structure similar to the physical object captured in section 5.4.1. Sensor viewpoints of these models are simulated in order to create point clouds representative of the field of view of the simulated sensor. Doing so affords ground truth

view alignment that is unobtainable when considering real world data of this nature. Point measurement samples are created such that the synthetic model is contained in a 2 unit cube ($[-1 \dots 1]$) bounding box and viewpoint densities are constructed to simulate (down-sampled) contemporary structured light scanners, containing ~ 5000 depth samples per view. Points per view are around $1/4$ of the point count from a typical Kinect point cloud of the physical tridecahedron thus keeping the repeated trial experimental setup, investigated in this section, feasible whilst maintaining a challenging data set making use of $\sim 1,250,000$ points in total. Additionally, in order to illustrate the capability of the proposed system to tackle extremely large realistic datasets, typical of modern depth sensors, we generate additional up-sampled versions of the previously explored synthetic datasets (section 3.5.1) containing ~ 50000 points per viewpoint, resulting in over 12 million points per dataset when using 250 viewpoints (see Tables 10, 11 in section 5.5 for dataset statistics and timing results).

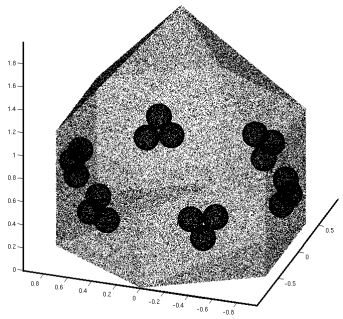
By generating a unit vector (representing a view) and extracting all point samples lying within the synthetic viewing frustum, an inexpensive method of building viewpoint specific point clouds is provided (refer to Figure 14 for a visualisation of this frustum technique). More costly viewpoint simulation alternatives involve performing ray-tracing to determine which object points are visible from a simulated sensor viewpoint or collecting points belonging to all front facing triangles. These alternatives provide a more realistic viewpoint simulation, potentially increasing the realism of the simulated data sets, especially if *e.g.* highly concave or self-occluding models are considered. However, in the experimental work detailed here we deal with convex models and decide to utilise the simple simulated frustum approach outlined, aiding sampling and view-set creation speed. Exploring the benefits afforded by a more advanced synthetic view generation process provides a further avenue for future work.

In summary, synthetic data are generated in a similar fashion to those described in chapter 3, but here we extend the number of simulated viewpoints to simulate a contemporary high-frame rate depth sensor resulting in hundreds of simulated point clouds per object model. Synthetic data sets begin in a perfectly aligned ground truth configuration. We perturb each view via applying a random rototranslation matrix (composed of a rotation and translation for each of the three axes, drawn uniformly at random) to get an initial alignment configuration. See Figure 56. It can be observed

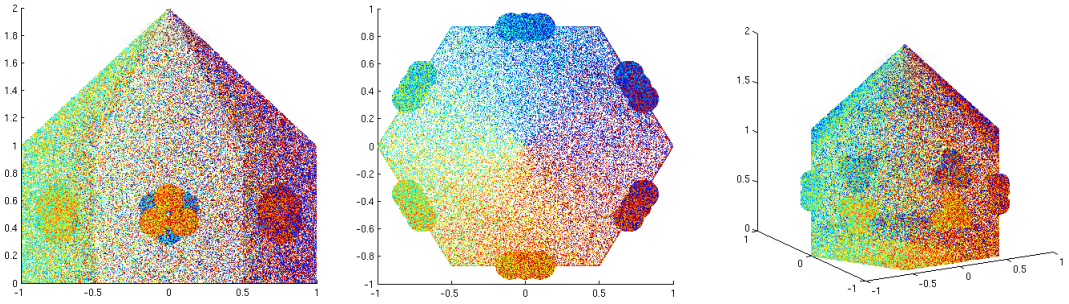
how point (x, y, z) measurements have been unit scaled in $[-1, 1]$, allowing for ease of familiarisation with the levels of random coarse seed transform strength applied.



(a) A synthetic tridecahedron model, built to approximate the physical object utilised in section 5.4.1.



(b) Point samples from the synthetic model. See text for additional detail.



(c) Depth measurements are collected in point clouds by sampling the model surface area that lies within each synthetic camera's frustum. Here one colour per point cloud is shown with views in a ground truth (perfect) alignment and zero simulated sensor measurement noise ($\mu = 0$, $\sigma = 0$). Front-to-parallel, top-down and real-world-sensor-approximation viewpoints of 250 point clouds. Best viewed in colour.

Figure 56

Starting with ground truth alignments and perturbing all viewpoints with random spatial nudges in this fashion is a suitable component of the experimental setup as this provides a neutral and flexible performance assessment method. With this data, we have the added advantage of possessing ground truth alignment since we have the original model. A similar strategy is recently used in [27] during their multi-view registration performance analysis. This synthetic perturbation tactic allows for the realistic simulation of a generic coarse alignment, contributing to an experimental setup that is not constrained to use a specific coarse alignment method with the possible bias that

this might introduce. The related problem of potentially influencing the reliability and repeatability of the results is avoided. The chosen method allows modulation of the starting distance from the desired optimal ground truth alignment. This in turn proves a useful and enabling feature for the stress testing we go on to explore in section 5.4.2.1. Varying misalignment scenarios can be built by randomly applying a bounded amount of angular and translation offset to each axis (per point cloud).

5.4.2.1 *Synthetic data: multi-view registration*

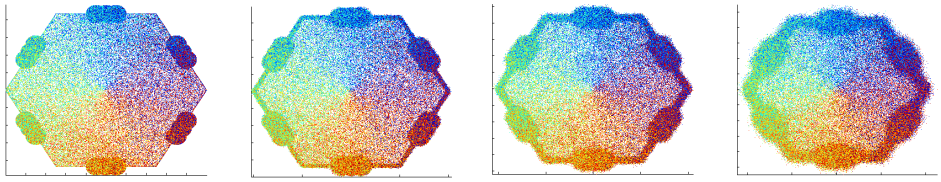
In this section we assess and compare the convergence properties of the proposed registration algorithm and those of [210], [262] using large synthetic view sets with known ground truth alignment. By increasing the magnitude of both the initial view misalignment and level of simulated sensor noise in the described synthetic data sets we provide a test bed to explore algorithm robustness when tasked with registering hundreds of views. It is noted that these synthetically generated coarse alignment scenarios may not fully satisfy our good initial alignment assumptions, since the aim here is to test the limits of our registration framework in order to widely assess the basin of convergence.

We consider the data set *synthetic tridecahedron* and randomly misalign each of the constituent point clouds and add Gaussian noise to point samples in order to simulate depth sensor sampling error.

View misalignment seeding is achieved by applying an independent, random, bounded amount of angular and translational offset to each of the three axes, for each viewpoint. This constitutes a strategy commonly undertaken (*e.g.* [27]) when performing fine registration stress testing. By increasing the range of random offset applied, 4 different levels of misalignment strength are considered in the coarse seeding scenarios. Translational offsets are drawn uniformly randomly, for each viewpoint, from the ranges $\{0, [-0.175 \dots 0.175], [-0.350 \dots 0.350], [-0.5 \dots 0.5]\}$ for weakest to strongest misalignment scenarios respectively. These levels of translational offset t are paired with random rotations drawn uniformly from similar ranges r to define random angular rotational misalignment (radians). We once again select parameter value ranges to represent coarse alignment configurations that might be achieved manually across a spectrum of novice to expert users, with the strongest misalignment providing more challenging seeding positions than those ever expected by any manual coarse alignment. These (r, t) pairings provide 4 coarse misalignment levels:

$$(r,t) \in \{(0,0), ([-0.175 \dots 0.175],[-0.175 \dots 0.175]), \dots, ([-0.5 \dots 0.5],[-0.5 \dots 0.5])\}$$

Additionally a varying level of 3D Gaussian noise with mean 0 and $\sigma \in \{0, 0.01, 0.02, 0.04\}$ is added to *each* dimension of *each* point sample in *each* point cloud to simulate depth sensor sampling error (see Figure 57). We note that simulating noise in this fashion employs equal variance in each spatial dimension (x, y, z) and this is considered likely to be an oversimplification of true measurement noise distributions exhibited by the real sensors (*e.g.* Kinect) utilised in this work. We leave exploring more advanced noise models to future work. The 4×4 levels of misalignment and added sensor noise considered result in the creation of 16 view set combinations with varying levels of coarse misalignment and simulated sensor noise.



(a) $\sigma = 0$ (b) $\sigma = 0.01$ (c) $\sigma = 0.02$ (d) $\sigma = 0.04$

Figure 57: Synthetic tridecahedron data set in ground truth alignment. Each subfigure displays 250 point clouds with increasing levels of Gaussian noise to simulate depth sensor sampling error. Sensor noise is simulated with equal variance in each spatial dimension (x, y, z). One colour per point cloud (best viewed in colour).

The KDE registration, Scanalyze [210] and Procrustes [262] techniques were applied to each of the constructed misaligned and noisy view-pose sets. Both the KDE registration and Procrustes techniques were iterated to error metric convergence and the Scanalyze method [227] was initiated using pairwise point-to-point matches between overlapping connected view subgroups before the global optimisation (proposed by Pulli [210]) was applied.

Figure 58 summarises resulting inter-point distance error metric μ_{ipd} values (defined in section 3.5.1.1) across all three explored methods using a single trial for each of the considered transform and noise parameter settings. Varying the magnitude of random coarse seed alignment constitutes a robustness test and results obtained suggest that the KDE registration technique is more robust than the competing techniques in avoiding

local minima, evidenced by the resilience of the former (and sensitivity of the others) to various levels of simulated misalignment.

The superior convergence properties of the KDE approach can be motivated by the fact that it optimally aligns all the views simultaneously at each iteration; on the contrary Scanalyze tries to optimally align each view with respect to the rest, in a sequential way. Although Scanalyze is computationally frugal (*c.f.* Table 11), the approach is liable to error propagation and the loop closure phenomena. In particular for the experiment in question we highlight gross failure results obtained by the Scanalyze method for the cases of $\mathcal{R}, \mathcal{T} = 0.0$ coarse misalignment (all noise levels) and $\mathcal{R}, \mathcal{T} = 0.175$, 0.5 coarse misalignments for sensor noise level $\sigma = 0.0$. These trials exhibit output with μ_{ipd} substantially larger than the input seed configuration. On investigation, what happens in practice here is that the algorithm runs into loop closure problems that propagate through a sequence of viewpoints leading to gross visual alignment failures and large corresponding quantitative error. Additionally we highlight poor registration outcomes for the proposed KDE method that deviate from the typically very promising results; notably the cases of coarse misalignments $\mathcal{R}, \mathcal{T} = 0.35$ and 0.5 combined with noise level $\sigma = 0.0$ along with $\mathcal{R}, \mathcal{T} = 0.5$ where $\sigma = 0.01$. These trial instances can be seen to have error metric values that are clearly worse than ground truth registration values and on inspection of the related qualitative results (see Figures 61e, 61f, 62f) it can be seen that *view-clique* registration error has adversely affected the result. The correlation between dissatisfying quantitative and qualitative results provides reassurance that the error metric made use of is sensible and in particular these clique based errors may be resolvable using our method by re-registering with larger bandwidth kernels.

Additionally the Procrustes based method of [262] can be seen to optimise local scan misalignment with respect to point pair distance at the expense of global object shape. Figure 59 additionally provides direct visualised comparison of the converged error landscape between the introduced method and [262] for the explored seed scenario combinations. As might be expected mean inter-point distance increases with both misalignment strength and simulated sensor noise in each case. The error surface generated by our registration approach, using the examined misalignment scenarios, exists below the Procrustes (Toldo et al. [262]) error surface in each examined instance (lower error) and, as error increases with simulated sensor noise, a modestly more elegant degradation can be observed in the case of KDE registration. Additionally, by accounting for

global object shape, the introduced method is seen to consistently reproduce object shape and structure in accordance with the ground truth.

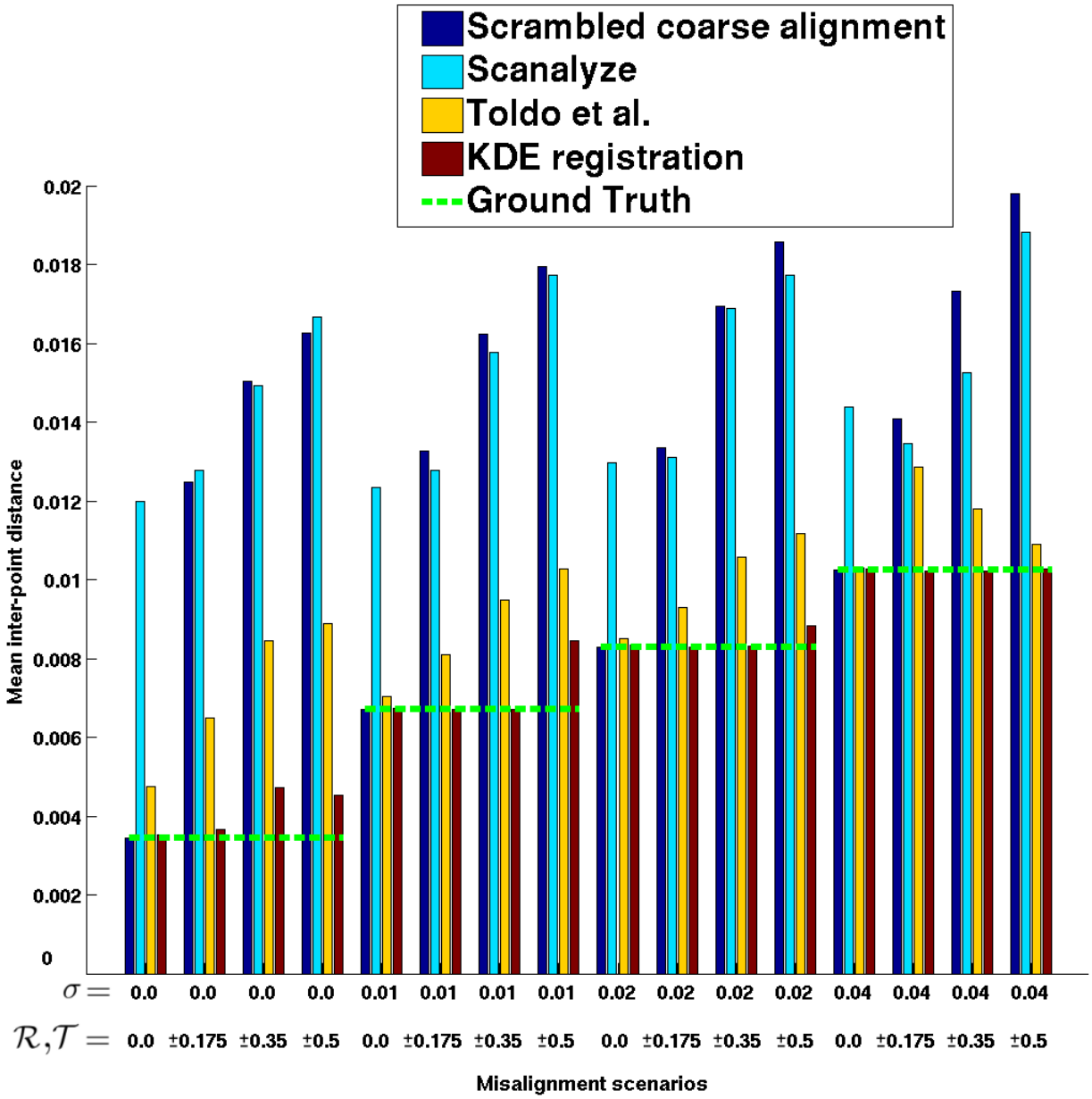


Figure 58: Robustness test: measured mean inter-point distance in post-registration view-set configurations. Three multi-view registration methods are evaluated across four levels of random coarse misalignment $\pm\{0.0, 0.175, 0.350, 0.5\}$ in combination with four levels of simulated sensor noise $\sigma \in \{0.0, 0.01, 0.02, 0.04\}$ (a single trial for each parameter combination).

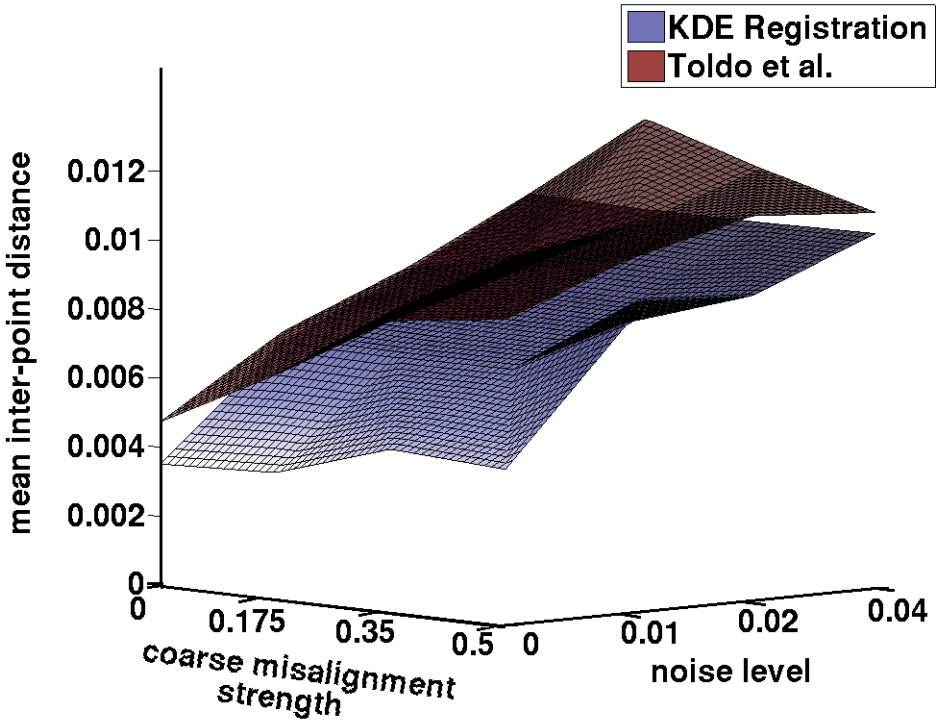
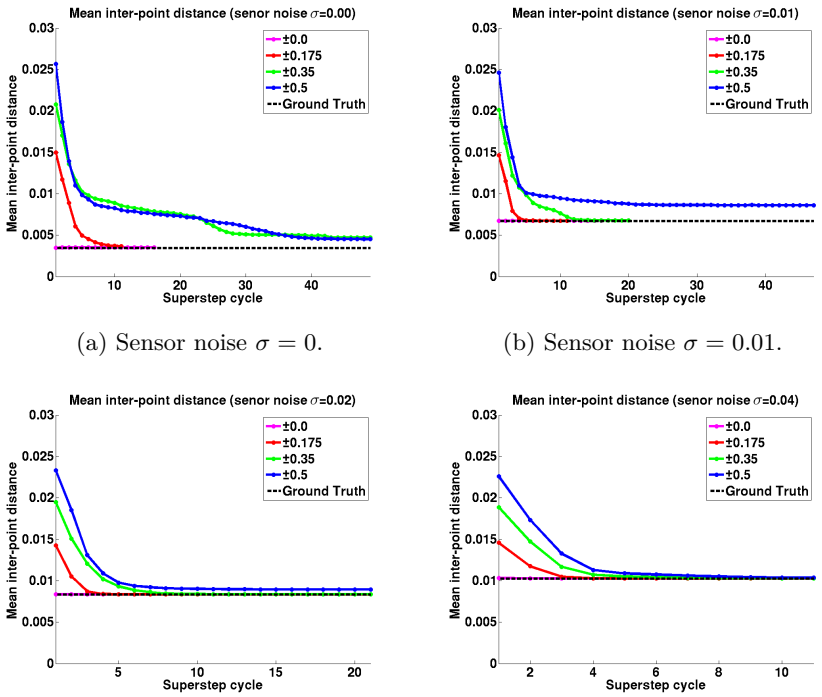


Figure 59: Robustness comparison: mean inter-point distance in post-registration view-sets. Error surfaces are visualised for the evaluated multi-view registration methods tested across four levels of random coarse misalignment $\pm\{0.0, 0.175, 0.350, 0.5\}$ in combination with four levels of simulated sensor noise $\sigma \in \{0.0, 0.01, 0.02, 0.04\}$. See text for discussion.

Additionally Figures 61 - 64, common subfigure sets (a)-(c), visualise the seed configurations for the misaligned data sets in the investigated combinations of simulated sensor noise and increasingly perturbed coarse misalignment. The corresponding subfigure sets (d)-(f) represent the registration obtained after our KDE registration is applied. The initial view scrambling becomes visually evident across the range of explored coarse alignment stress levels[3]. Final KDE registration results are in general visually close to their respective ground truth. An exception is noted in the case of Figure 61e where a failure mode for the scenario involving random transforms drawn from ± 0.350 and noise level $\sigma = 0$ can be observed. In the noted case a small collection of overlapping views find a local minima involving an incorrectly aligned view-clique in the lower-right quadrant of the view-set. It is thought that this is likely caused by the

3 Point cloud visualisation results not shown for the trivial cases involving misalignment seeds drawn from random transforms of size $(0, 0)$. Starting in the ground truth positions, these experiments effectively converge immediately (as confirmed in Figure 60).

particular random seed transforms, drawn in this problem instance, proving too great for the surface density estimation smoothing to overcome. However, all remaining trials at this level of random scrambling (yet displaying higher sampling noise, Figures 62e, 63e, 64e) are able to converge to visually satisfying results when compared with their respective ground truth (Figure 57). As the level of misalignment is increased to ± 0.500 further minor local minima occur (see Figures 61f, 62f) however even under these large transform offsets (and high noise levels) successful registration, producing alignments visually similar to the ground truth (Figures 63f, 64f) is still achievable from input that can be considered far beyond that defined as a reasonable coarse alignment. Figure 60 presents the mean inter-point distance (μ_{ipd}) error metric convergence for the explored coarse alignment and noise level combinations using our KDE registration strategy. The μ_{ipd} is also measured in the ground truth pose alignment, for each explored noise level, providing a straight forward quantitative assessment of the registration quality in each case. Algorithm termination is in this case defined by error metric convergence and quantitative comparison between registration techniques is again collated in Tables 10 and 11.



(a) Sensor noise $\sigma = 0$.

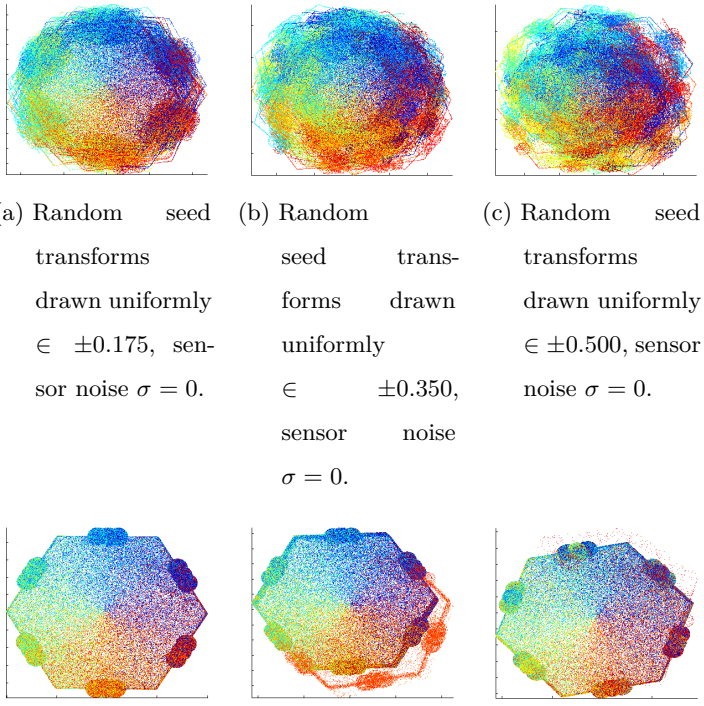
(b) Sensor noise $\sigma = 0.01$.

(c) Sensor noise $\sigma = 0.02$.

(d) Sensor noise $\sigma = 0.04$.

Figure 60: Mean inter-point distance error metric evolution per superstep iteration. Mean inter-point distance, measured in the ground truth configuration, for the investigated noise levels also shown. Random seed transforms drawn uniformly $\in \pm\{0, 0.175, 0.350, 0.5\}$.

The demonstrated robustness against misalignment and sensor noise, observed with the data sets utilised throughout the chapter, reveal that the range of seed configuration, handled by our registration approach can be considered wide with respect to what are commonly intended and required as "good" initial alignment conditions. The experimental work illustrates that what might be typically regarded as a "good" initial coarse alignment can be regarded as conservative in our framework. This is evidenced by the demonstrated ability to produce reasonable results even when undertaking the heaviest levels of misalignment and noise tested.



(a) Random seed transforms drawn uniformly $\in \pm 0.175$, sensor noise $\sigma = 0$.

(b) Random seed transforms drawn uniformly $\in \pm 0.350$, sensor noise $\sigma = 0$.

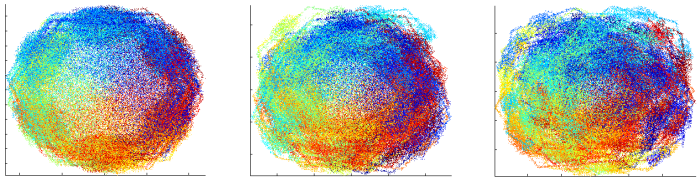
(c) Random seed transforms drawn uniformly $\in \pm 0.500$, sensor noise $\sigma = 0$.

(d) Post KDE registration from $\in \pm 0.175$ coarse seed, sensor noise $\sigma = 0$.

(e) Post KDE registration from $\in \pm 0.350$ coarse seed, sensor noise $\sigma = 0$.

(f) Post KDE registration from $\in \pm 0.500$ coarse seed, sensor noise $\sigma = 0$.

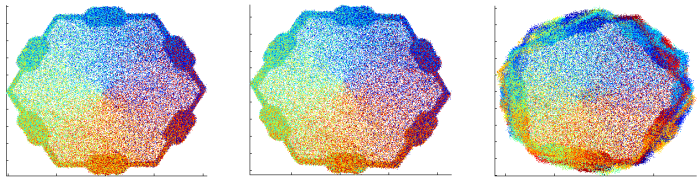
Figure 61: Top row: tridecahedron with $\sigma = 0$ sampling noise and seed positions for differing levels of coarse misalignment. Bottom row: corresponding KDE registration results.



(a) Random seed transforms drawn uniformly $\in \pm 0.175$, sensor noise $\sigma = 0.01$.

(b) Random seed transforms drawn uniformly $\in \pm 0.350$, sensor noise $\sigma = 0.01$.

(c) Random seed transforms drawn uniformly $\in \pm 0.500$, sensor noise $\sigma = 0.01$.

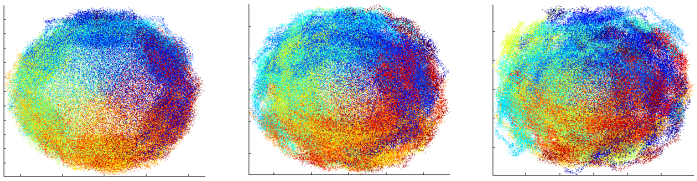


(d) Post KDE registration from $\in \pm 0.175$ coarse seed, sensor noise $\sigma = 0.01$.

(e) Post KDE registration from $\in \pm 0.350$ coarse seed, sensor noise $\sigma = 0.01$.

(f) Post KDE registration from $\in \pm 0.500$ coarse seed, sensor noise $\sigma = 0.01$.

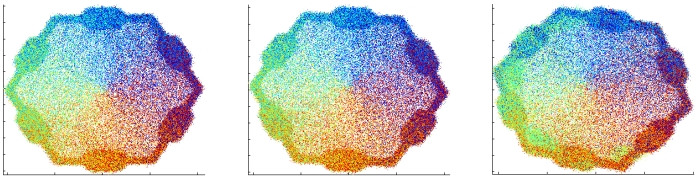
Figure 62: Top row: tridecahedron with $\sigma = 0.01$ sampling noise and seed positions for differing levels of coarse misalignment. Bottom row: corresponding KDE registration results.



(a) Random seed transforms drawn uniformly $\in \pm 0.175$, sensor noise $\sigma = 0.02$.

(b) Random seed transforms drawn uniformly $\in \pm 0.350$, sensor noise $\sigma = 0.02$.

(c) Random seed transforms drawn uniformly $\in \pm 0.500$, sensor noise $\sigma = 0.02$.

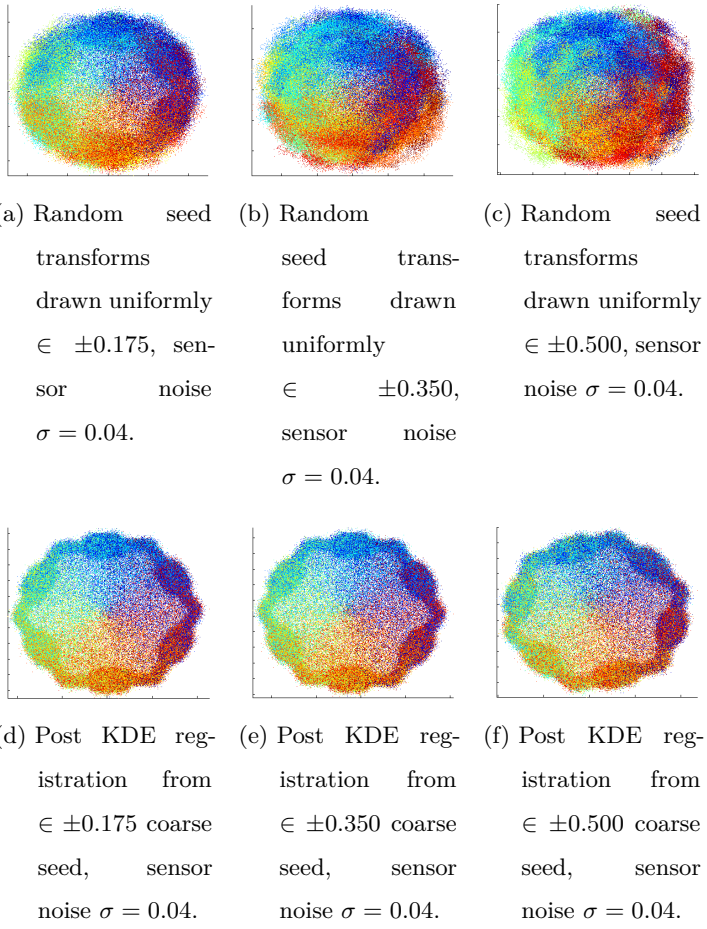


(d) Post KDE registration from $\in \pm 0.175$ coarse seed, sensor noise $\sigma = 0.02$.

(e) Post KDE registration from $\in \pm 0.350$ coarse seed, sensor noise $\sigma = 0.02$.

(f) Post KDE registration from $\in \pm 0.500$ coarse seed, sensor noise $\sigma = 0.02$.

Figure 63: Top row: tridecahedron with $\sigma = 0.02$ sampling noise and seed positions for differing levels of coarse misalignment. Bottom row: corresponding KDE registration results.



(a) Random seed transforms drawn uniformly $\in \pm 0.175$, sensor noise $\sigma = 0.04$.

(b) Random seed transforms drawn uniformly $\in \pm 0.350$, sensor noise $\sigma = 0.04$.

(c) Random seed transforms drawn uniformly $\in \pm 0.500$, sensor noise $\sigma = 0.04$.

(d) Post KDE registration from $\in \pm 0.175$ coarse seed, sensor noise $\sigma = 0.04$.

(e) Post KDE registration from $\in \pm 0.350$ coarse seed, sensor noise $\sigma = 0.04$.

(f) Post KDE registration from $\in \pm 0.500$ coarse seed, sensor noise $\sigma = 0.04$.

Figure 64: Top row: tridecahedron with $\sigma = 0.04$ sampling noise and seed positions for differing levels of coarse misalignment. Bottom row: corresponding KDE registration results.

5.4.2.2 *Synthetic data: view-set influence on model fitting*

In addition to comparing robustness and registration quality on synthetic data sets, model fitting is performed to provide further quantitative evaluation. The objective explored involves discerning effects on model building quality in relation to increasing view-set magnitude (and therefore redundant sampling per physical object location). We design the following experimental setup to test if reconstruction quality can be improved by increasing the magnitude of registered data sets. To experimentally test *hypothesis* 3 (see section 5.1) we are interested in the model fitting error produced by a standard RANSAC [91] model-fitting algorithm when applied to spheres extracted from view-set KDE registration output (produced from ± 0.175 coarse alignment seed transform, $\sigma = 0.01$ sensor noise input). Utilising synthetic data sets again gives the

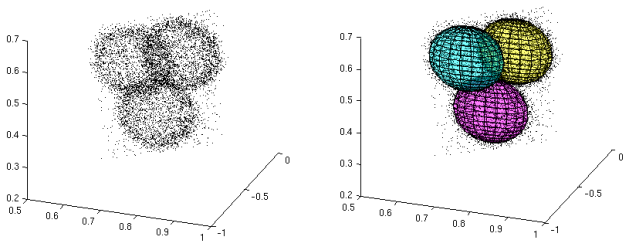
ground truth for various geometrical properties, obtainable from the original point sampled model (refer to Figure 56a).

Sphere fitting to range data is a common task (*e.g.* [99]) and, in an attempt to mimic the physical object explored previously in section 5.4.1, here we utilise synthetic models that contain sets of three spheres (*tri-sphere* features) placed on each vertical planar face of the generated tridecahedron. Ground truth statistics are collected from the synthetic model including true synthetic sphere centroid locations and true (uniform) sphere radii length. Depth points from converged registered view-sets that sample the tri-sphere features are manually segmented from the sets and provided as input to a RANSAC [91] fitting algorithm to compare the quality of the resulting fitted sphere statistics. The stated hypothesis is tested by varying the *number of views*, from the set of registered views, that contribute samples to the amalgamated regions containing the tri-sphere features.

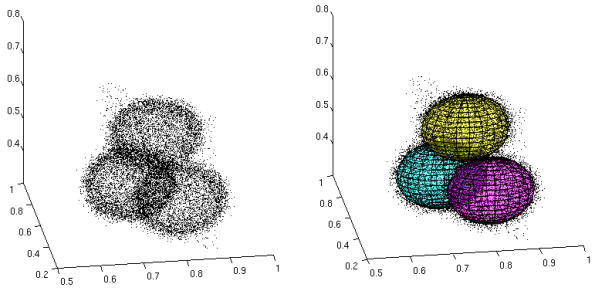
The RANSAC sphere fitting algorithm is tasked with finding the *three best fitting spheres* in the provided tri-sphere region point set (best sphere-fits defined by a standard SSD fitting error). A greedy RANSAC strategy is employed such that once a sphere is found in the provided tri-sphere region, inliers are removed from the point set and the RANSAC sphere-fit repeated using the remaining points. This process continues until three spheres have been fitted (or the maximum number of RANSAC trials reached).

Post-registration view-sets are integrated such that all samples (from each view) form a single large, converged, point set. This combined set is easily manually segmented to retrieve points extracted from tri-sphere feature regions. An example set of post-registration, manually segmented viewpoints containing samples contributing to tri-sphere regions (and the RANSAC fits found) are provided in Figure 65.

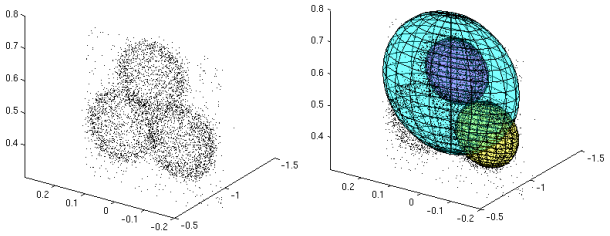
Manual segmentation of the tri-sphere feature regions from the amalgamated set of points reveal, as expected, that due to the angular positioning of the synthetic depth sensor locations (uniformly distributed in the viewing sphere) each region is visible from a subset of the original point clouds (on average a feature is "visible" from ~ 25 point clouds). By artificially restricting the number of viewpoints that contribute to each segmented feature region (via random selection) and performing RANSAC fitting on a range of these restricted sets, we explore the effect that increasing the number of registered views (redundant sampling per physical location) has on model fitting accuracy.



(a) 4 registered, amalgamated and segmented viewpoints (± 0.175 coarse rototranslation seed, noise $\sigma = 0.01$) forming a point set containing a tri-sphere region. Example successful fit with mean radius$_\mu = 0.10087$.



(b) 8 registered, amalgamated and segmented viewpoints (± 0.175 coarse rototranslation seed, noise $\sigma = 0.01$) forming a point set containing a tri-sphere region. Example successful fit with mean radius$_\mu = 0.10066$.



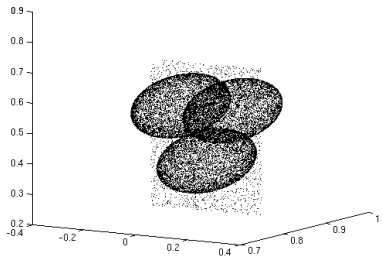
(c) 4 registered, amalgamated and segmented viewpoints (± 0.175 coarse rototranslation seed, noise $\sigma = 0.01$) forming a point set containing a tri-sphere region. Example RANSAC fit failure mode with mean radius$_\mu = 0.13914$.

Figure 65: Post-registration point clouds amalgamated to form integrated point sets. Left column exhibits manually segmented view sets which allow *tri-sphere* feature regions to be extracted. A RANSAC fit (right column) is applied to these point sets and sphere fits plotted. When view count is doubled an improved fit is found yet gross failure modes are also possible at the (relatively low) $\sigma = 0.01$ noise level.

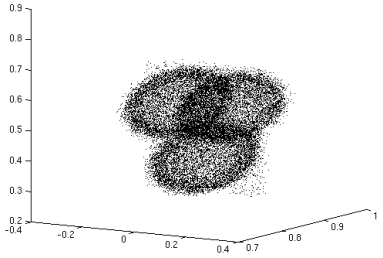
5.4.2.3 *Synthetic data: Model fitting with view-set restriction*

Tridecahedron view-sets seeded with ± 0.175 magnitude coarse misalignments are again considered. Registration results were previously shown to be visually consistent and satisfying, with respect to ground truth, across considered simulated sensor noise levels $\sigma \in \{0, 0.01, 0.02, 0.04\}$ for this magnitude of coarse misalignment (section 5.4.2.1).

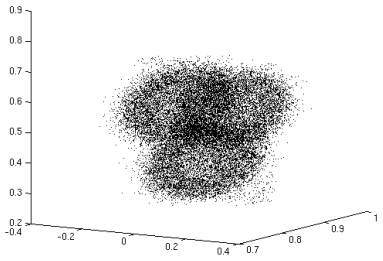
Taking these registered view-sets and defining amalgamated point cloud regions by manually segmenting all 6 tri-sphere features (from each vertical planar side of the registered view-set) results in input suitable for the described RANSAC sphere fitting process (input examples found in Figure 66). The RANSAC process computes centroid locations and radii lengths for each fitted sphere model found. By comparing (1) fitted sphere radii values to the ground truth model sphere radius and (2) the RMS distance between fitted sphere centroids and *true* model sphere centroids we provide simple quantitative metrics to assert how well the RANSAC tri-sphere model fitting process can be accomplished when a varying number of viewpoints are allowed to contribute point sample information to the amalgamated point set regions.



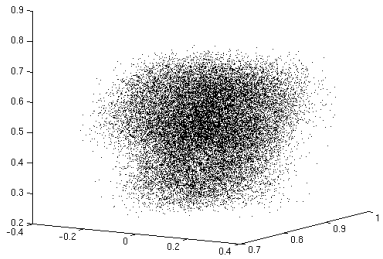
(a) Synthetic sampling noise $\sigma = 0$, example point sample region constructed from 4 viewpoints (point clouds).



(b) Synthetic sampling noise $\sigma = 0.01$, example point sample region constructed from 4 viewpoints (point clouds).



(c) Synthetic sampling noise $\sigma = 0.02$, example point sample region constructed from 5 viewpoints (point clouds).



(d) Synthetic sampling noise $\sigma = 0.04$, example point sample region constructed from 6 viewpoints (point clouds).

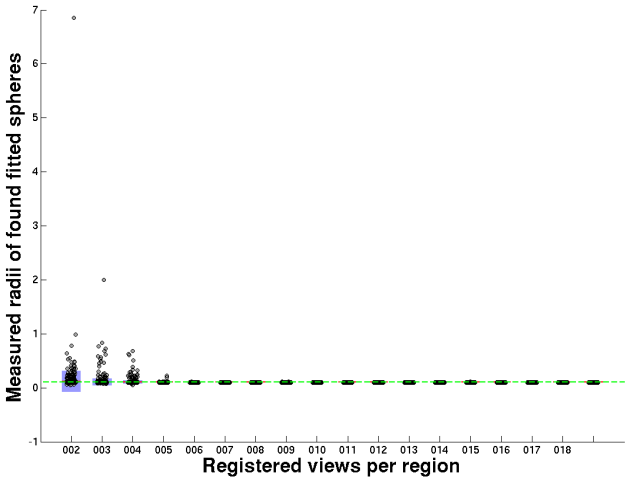
Figure 66: Amalgamated tri-sphere segmented regions for various simulated sensor noise levels. Each amalgamated region contains (in this instance) $4-6$ registered viewpoints contributing points to the integrated point set. Individual spheres can be seen to become visually harder to discern as simulated noise increases.

Each individual RANSAC sphere fit attempt is limited to 10000 trials (per sphere search) and the overarching sphere-fitting process attempts to find 18 spheres per segmented view-set: one tri-sphere (3 spheres) per side for each of the 6 tridecahedron vertical planes. This experimental process (attempting to fit 18 spheres) is repeated 100 times for each of the four investigated view-set noise levels in order to examine RANSAC fitting variance. For each of three simulated noise levels $\sigma \in \{0, 0.01, 0.02\}$ registered views are amalgamated and viewpoints, where tri-sphere regions are "visible", are selected by segmentation. These tri-sphere regions can then be reconstructed

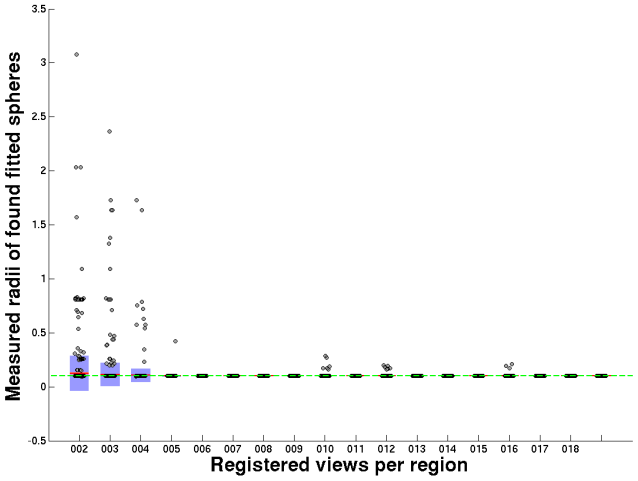
by varying the contributing viewpoint count. Randomly including or excluding views in which each tri-sphere region is "visible" allows the number of viewpoints providing sphere feature information to be artificially determined.

Example: If a tri-sphere region is found to be "visible" in 20 registered point clouds in total, evaluating RANSAC fitting using an amalgamated point set, contributed to by *e.g.* 10 views, can be performed by randomly selecting 10 views from the original 20 in which the tri-sphere feature is visible.

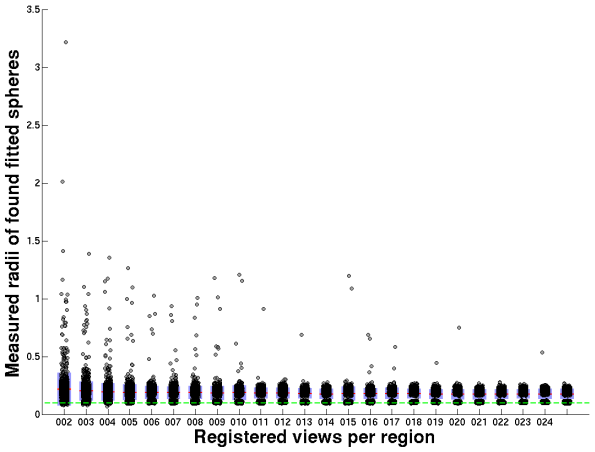
RANSAC sphere fitting is performed on each tri-sphere region utilising amalgamated point sample information whilst varying the number of contributing viewpoints $(2-25)$. The tri-sphere radii and sphere centroids found by the RANSAC fit are compared to the known ground truth. Across 100 repeated RANSAC trials we potentially generate 100×18 fitted sphere radii and centroid values using amalgamated point samples built from viewpoint subsets of varying size (viewpoint subset sizes: $2-25$, for each of 4 sensor noise levels). We discard individual sphere-fit instances where RANSAC is unable to find a valid sphere in ≤ 10000 trials and plot results for $2-25$ amalgamated viewpoint sizes. Model reconstructions (containing 6 tri-sphere regions) attempt to fit spheres using a minimum viewpoint count of 2 and a maximum viewpoint count defined by depth sample viewpoint-memberships contributing to model feature regions (this exhibits small variance with simulated sensor noise level σ). The ground truth model sphere radius is 0.1 unit and resulting radii fits are found in Figure 67 for point sets exhibiting Gaussian sensor noise $\sigma \in \{0, 0.01, 0.02\}$ (with enlargement of the $\sigma = 0.01$ case found in Figure 68).



(a) $\sigma = 0$

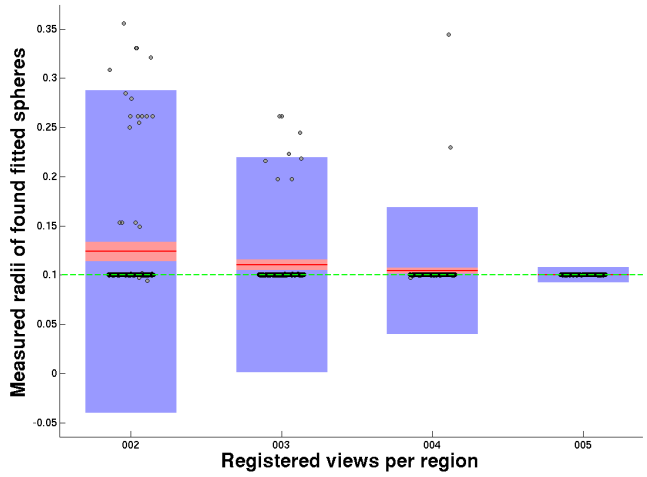


(b) $\sigma = 0.01$

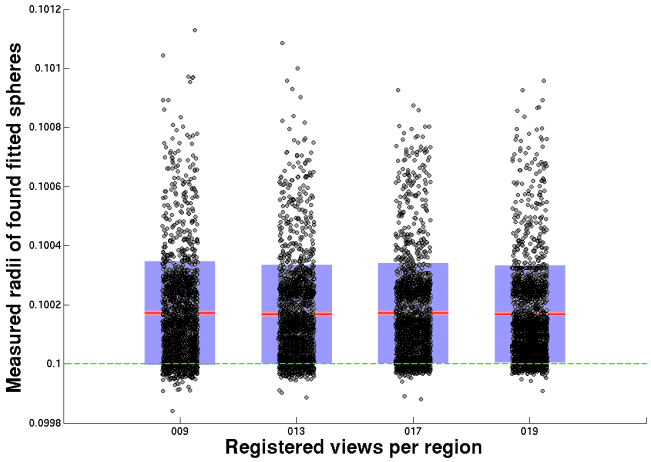


(c) $\sigma = 0.02$

Figure 67: RANSAC fitted sphere radii for synthetic registered point clouds with varying simulated noise level. Tri-sphere feature regions are defined from growing sets of registered point clouds per region (x-axis). Each column contains ~ 1800 radii from repeated sphere fit trials considering point sample information contributed to from increasingly large point cloud view collections. See text for further detail.



(a) Enlarged view of radii fitting results for views per region $002 - 005$ from Figure 67b. Horizontally jittered raw radii fits are displayed with mean values (red), 95% confidence intervals (pink), Standard Deviation (purple - see 67b for extreme outliers) and ground truth model sphere radius, 0.1 unit (green). See text for discussion.



(b) Enlarged view of radii fitting results for 009, 013, 017 and 019 views per region (built by considering contributions from $9 - 19$ registered viewpoints) from previous Figure 67b. Graph colouring as above (68a). With viewsets of this size, fitting error is consistently an order of magnitude below the simulated noise level.

Figure 68

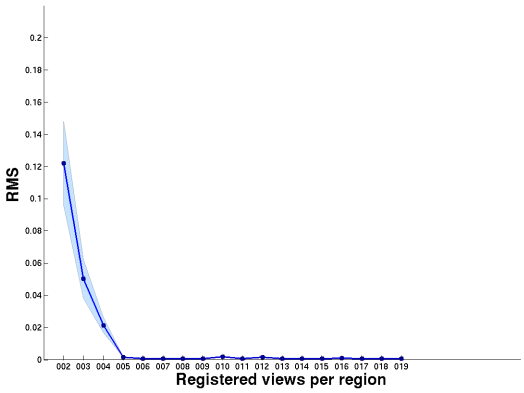
Tri-sphere feature regions are RANSAC fitted with spheres from growing sets of registered point clouds per region (Figure 68a, 68b x-axis). Each column in these figures contains $\sim$ 1800 found radii (horizontally-jittered to aid visual clarity) from repeated

sphere fit trials that consider point sample information contributed from increasingly large point cloud view collections. Raw radii fits are displayed with mean values (red), 95% confidence intervals (pink), Standard Deviation (purple) and the ground truth model sphere radius is 0.1 unit (green). It can be observed that as the contributing viewpoint count increases, fitting variance drops sharply and the mean fitted sphere radii length converges on the ground truth value. The effect size is small in the simple example implemented, however this provides initial evidence that larger view-sets can mitigate sensor noise and contribute to improving model fit quality. For the cases of $\sigma \in \{0, 0.01\}$ synthetic sensor noise we find experimentally that mean radii fit values asymptote when ≥ 5 views contribute to tri-sphere region information. Sphere fitting results for the zero synthetic noise case proves largely comparable to the $\sigma = 0.01$ case when enlarged (exhibiting convergence towards radii ground truth within ~ 5 contributing views) while, as might be expected, increasing the noise level ($\sigma = 0.02$) inhibits this convergence. Results are summarised in Tables 7-9. In addition to sphere radii fits we consider fitted sphere centroid locations and compare these to the model ground truth centroid locations. By greedily performing one-to-one closest point matching between each set of 18 fitted and true centroids (one tri-sphere on each of 6 model vertical planes) a simple RMS Euclidean distance metric and standard deviation is calculated for each (of 100) RANSAC trials, for each amalgamated view-set size, for each simulated noise level $\sigma \in \{0, 0.01, 0.02\}$.

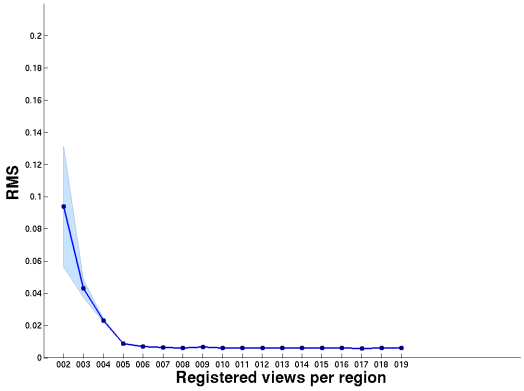
It is recognised that the maximum number of viewpoints contributing to each feature region increases marginally as synthetic noise grows stronger (*c.f.* Figures 69a, 69c). This is explained as depth samples (afforded by viewpoints that record surface area from planar object regions, as defined by the ground truth) can become spatially distorted by sensor noise to lie within tri-sphere regions. The probability of this occurring increases with sensor noise strength. The resulting (small) number of points (incorrectly) lying in segmented tri-sphere regions, allow these views to be considered to contain "visible" tri-sphere feature regions. Additionally, we omit registered view-sets exhibiting the largest considered noise level $\sigma = 0.04$ as the simple RANSAC algorithm was unable to consistently find spheres (of any size or location) in data exhibiting this level of simulated noise. It is concluded that (for the simple synthetic experimental setup constructed) we approach an upper bound on the level of noise that can be successfully mitigated by increasing view count. The $\sigma = 0.02$ strength, mean radii precision

and coinciding-centroid RMS both suffer significantly (*c.f.* ground truth). Furthermore spheres sampled at the $\sigma = 0.04$ noise level become increasingly difficult to perceive with the human eye (*c.f.* Figure 66d).

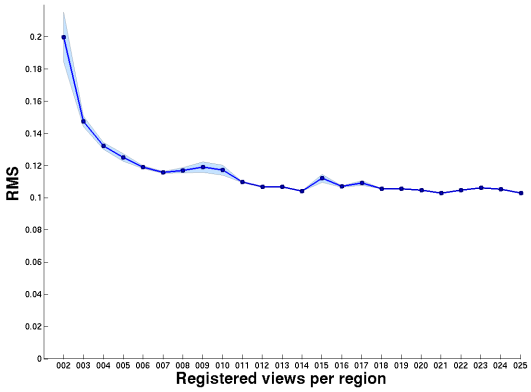
Intuitively the coinciding-centroid error metric is large when fitted and true centroid position pairs differ and approaches zero when centroid pairs coincide. The progression of the metric, as amalgamated point sets grow in size, are provided in Figure 69 for simulated sensor noise levels $\sigma \in \{0, 0.01, 0.02\}$.



(a) Simulated sensor noise level $\sigma = 0$.



(b) Simulated sensor noise level $\sigma = 0.01$.



(c) Simulated sensor noise level $\sigma = 0.02$.

Figure 69: Coinciding-centroid model-fit quality metric. As amalgamated view-set size increases (x-axis), mean values and standard deviation converge indicating agreement between model fit and ground truth. See text for further details. Input point clouds with simulated depth sensor noise $\sigma = \{0, 0.01, 0.02\}$ are considered in (a)-(c) respectively.

Table 7: RANSAC model fitting results for sensor noise $\sigma = 0$.

View points utilised for RANSAC fit	# Valid sphere fits found over 100 trials (max 1800)	Mean point samples per tri-sphere feature region	Mean fitted radius	Median fitted radius	Standard deviation of fitted radius	Mean centroid coincidence RMS over 100 trials	RMS centroid distance SD over 100 trials
2	1044	# points = 4467	0.115937	0.100166	0.13371	0.1220	0.0261
3	1656	# points = 8090	0.107416	0.100166	0.10804	0.0503	0.0120
4	1764	# points = 12098	0.104380	0.100148	0.06294	0.0215	0.0047
5	1800	# points = 16062	0.101351	0.100148	0.00424	0.0016	0.0008
6 . . . 13	1800	20126 $\leq$ # points $\leq$ 52361	$\leq$ 0.10053	$\leq$ 0.100152	$\leq$ 0.00218	$\leq$ 0.0019	$\leq$ 0.0006
14	1782	# points = 56451	0.099169	0.100144	0.00397	0.0006	0.0011
15 . . . 18	1800	60513 $\leq$ # points $\leq$ 72482	$\leq$ 0.1002	$\leq$ 0.100144	$\leq$ 0.00116	$\leq$ 0.0011	$\leq$ 0.0012
Ground truth	1800	∞	0.10	0.10	0	0	0

Table 8: RANSAC model fitting results for sensor noise $\sigma = 0.01$.

View points utilised for RANSAC fit	# Valid sphere fits found over 100 trials (max 1800)	Mean point samples per tri-sphere feature region	Mean fitted radius	Median fitted radius	Standard deviation of fitted radius	Mean centroid coincidence RMS over 100 trials	RMS centroid distance SD over 100 trials
2	1332	# points = 4304	0.124753	0.101917	0.16199	0.0939	0.0374
3	1746	# points = 8146	0.110436	0.101587	0.10930	0.0431	0.0055
4	1800	# points = 12059	0.104774	0.101624	0.06444	0.0231	0.0036
5	1800	# points = 16163	0.100721	0.101529	0.00761	0.0088	0.0011
6 . . . 18	1800	20096 $\leq$ # points $\leq$ 72776	$\leq$ 0.101549	0.101603	$\leq$ 0.0021	$\leq$ 0.0071	$\leq$ 0.0033
Ground truth	1800	∞	0.10	0.10	0	0	0

Table 9: RANSAC model fitting results for sensor noise $\sigma = 0.02$.

View points utilised for RANSAC fit	# Valid sphere fits found over 100 trials (max 1800)	Mean point samples per tri-sphere feature region	Mean fitted radius	Median fitted radius	Standard deviation of fitted radius	Mean centroid coincidence RMS over 100 trials	RMS centroid distance SD over 100 trials
2	1332	# points = 4346	0.216798	0.204350	0.14388	0.1999	0.0155
3	1710	# points = 8131	0.201330	0.201202	0.08374	0.1474	0.0033
4	1782	# points = 12001	0.191798	0.197552	0.07638	0.1322	0.0022
5	1800	# points = 15878	0.188483	0.197881	0.06884	0.1250	0.0024
6	1800	# points = 19939	0.179203	0.197189	0.05723	0.1191	0.0008
7	1800	# points = 23878	0.176871	0.198001	0.05538	0.1158	0.0005
8	1800	# points = 27812	0.177199	0.196152	0.05824	0.1171	0.0016
9	1800	# points = 31800	0.164475	$\leq$ 0.197637	0.06131	0.1190	0.0033
10 . . . 24	1800	36243 $\leq$ # points $\leq$ 95917	$\leq$ 0.1650	0.198118	$\leq$ 0.05146	$\leq$ 0.1172	$\leq$ 0.0031
Ground truth	1800	∞	0.10	0.10	0	0	0

The first column of Tables 7 - 9 provide the number of views (point clouds) utilised to form an amalgamated point set that is provided as input to the RANSAC sphere fitting process. Over 100 RANSAC trials, a maximum of $18 \times 100 = 1800$ spheres can potentially be found in the input point clouds and the success count for integrated point sets, built from the merged views, is provided in the second column. The remaining columns provide quantitative information on geometrical properties (radii, centroid separation) of the fitted spheres averaged across all RANSAC trials. In particular, the third and fourth columns report mean and standard deviation for the sphere radii obtained from the fitted spheres and columns five and six provide the mean and standard deviation of the coinciding-centroid model-fit. As the synthetic noise level increases, the accuracy of sphere model fitting predictably decreases however this effect can be observed to be mitigated by increasing view counts and redundant point sampling per location. Specifically in the cases of $\sigma = 0$ and $\sigma = 0.01$ noise levels asymptotic behaviour and convergence to ground truth model values are achieved by obtaining *c.* 5 views of the simple object feature regions. Stronger noise ($\sigma \geq 0.02$) prevents the simple model fitting approach from converging to ground truth values (using the explored viewpoint ranges). Reduction in error can however still be observed as view counts increase. This simple model fitting provides initial evidence in support for the hypothesis made in section 5.4.2.2; redundant point sampling has the ability to contribute meaningfully to mitigating sensor noise and improve model fitting quality.

5.4.2.4 *Synthetic data: summary*

The obtained synthetic registration results provide confirmation of the suitability of the coarse alignment pose initialisation method. Additionally the observed stability of the registration results (shown in Figures 59, 58), irrespective of the random seed configuration, provides indirect confirmation that the coarsely aligned seed positions are found within the basin of convergence of the method. A rigorous mathematical definition and assurance about the basin of convergence of the proposed approach is not offered here (many possible factors influence the convergence basin shape and dimension). The experimental work does however provide evidence that the KDE registration approach is capable of handling up to moderate misalignments (*i.e.* unfavourable starting conditions that exceed those which might be normally expected as input for a global registration phase) when applied to large numbers of view-sets. Importantly error functions, of qual-

ity metrics utilised, are shown to consistently and regularly progress towards a global ground truth minimum.

Additionally, experimental evidence provides support for the claim that high view-set magnitude (and associated redundant point sampling) can contribute to mitigating sensor sampling noise effects and therefore improve model fitting (*e.g.* surface reconstruction) accuracy and quality. An obvious direction for future work involves repeating these initial synthetic experiments with state-of-the-art depth sensor data to explore if similar advantages can be gained when attempting to mitigate real-world sensor noise distributions. In the following experimental sections we evaluate further data sets from additional sources and real-world depth sensors, exploring the ability of the proposed method to handle large view-sets of distinct, complex and varied object shape.

5.4.3 *Stuttgart range images: data sets*

The Stuttgart range image database ([214], [123]) is a range image resource containing various object models from which large numbers of depth images have been produced per model. The database provides further suitable data sets for the experimental work carried out in this chapter. The resource consists of collections of range images obtained from 42 high-resolution polygonal models that are obtained both from existing synthetic object models found on the world wide web and additionally, models created by laser scanning physical objects (carried out in the Stuttgart lab). Synthetic range images of object models are generated by varying synthetic-camera viewing angles and positions in relation to the object and creating range images by sampling the visible surface in accordance with the synthetic-camera line of sight. The resolution of each resulting range image is 400×400 pixels with a single measurement value at each pixel (distance from the synthetic sensor). This potentially gives 160000 depth samples per image and typically results in $\sim 5000 - 50000$ valid 3D points per viewpoint (once non-object depth image pixels, containing NaN values, have been removed). By varying the spatial step size and viewing angle between synthetic-camera sensor positions, data sets of many viewpoints are created for each object model. The database offers sets of 258 range images per object model which we use to reproject point clouds of each view to 3D space using a standard pinhole camera model. Example synthetic depth images

from this resource (*42_fighter* object) are provided in Figure 70 and examples of point clouds, produced by reprojecting the depth images, are presented in Figure 71.

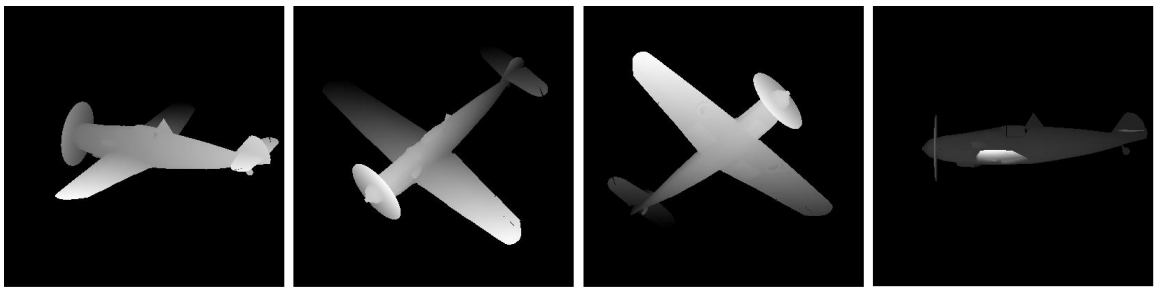


Figure 70: Example range images from the Stuttgart [214] range image database. Each object model is used to generate 66 or 258 range images. Object viewing angles differ by $23-26^{\circ}$ degrees (example images from the *42_fighter* object).

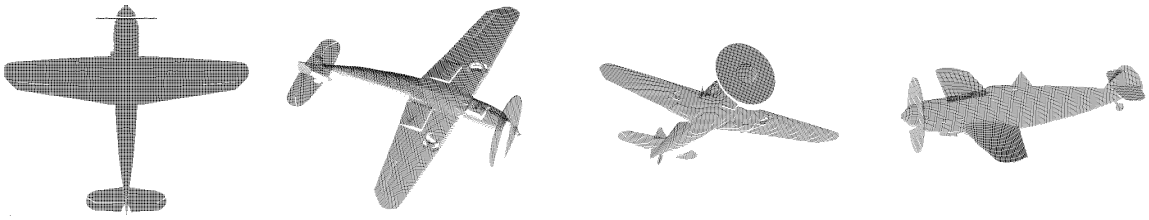
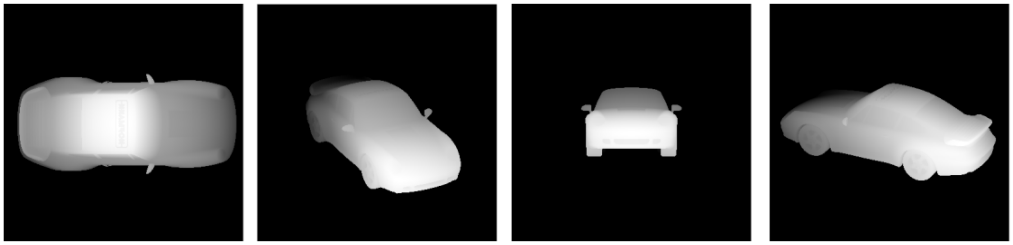


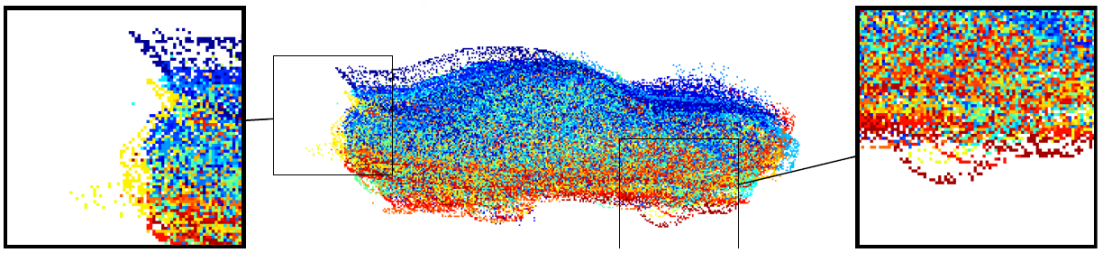
Figure 71: Example reprojected point clouds produced using a range image data set from the Stuttgart DB [214]. A pinhole camera model is used to generate a point cloud from each depth image that are then provided as input to the compared multi-view registration algorithms. Here example point clouds are reprojected from range images using the *42_fighter* data set (see Figure 70).

5.4.3.1 *Stuttgart range images: multi-view registration*

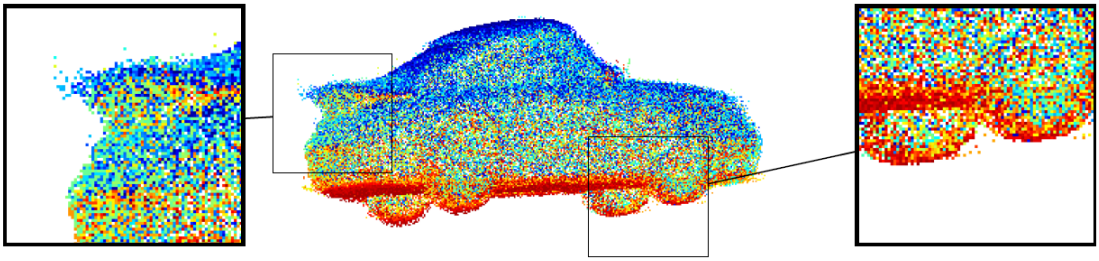
Once the Stuttgart range image sets have been reprojected to point clouds, views are firstly brought into a common frame of reference (coarse alignment) using the simple feature based coarse alignment strategy outlined in section 5.2.1. For the Stuttgart data sets, ground truth registration is not made use of (unavailable). Any large failure modes in the resulting automated coarse alignment are mitigated using additional manual hand-alignment at this stage (manual alignment intervention was typically required for $0-40\%$ of considered viewpoints per Stuttgart data set). This coarse alignment strategy generates view configurations that can be considered visually similar to the validated coarse alignment configurations constructed synthetically in section 5.4.2.1. See Figures 72b and 73a for examples of Stuttgart data sets in coarse alignment configurations.



(a) Example range images from the Stuttgart *17_porsche* image set.

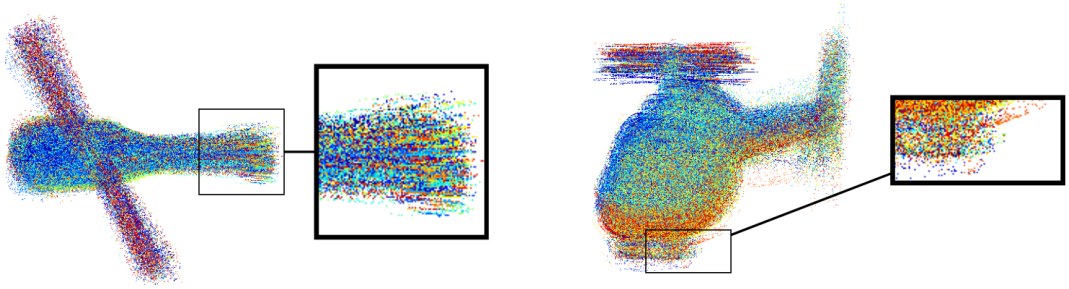


(b) Stuttgart *17_porsche* data set after spin image coarse registration.

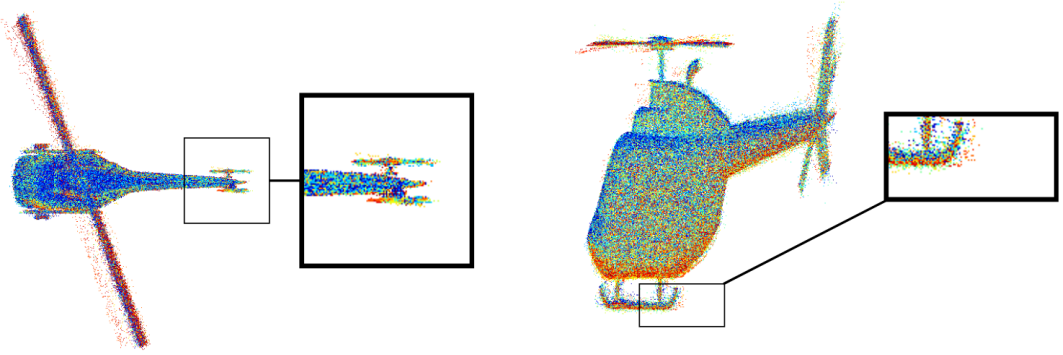


(c) Stuttgart *17_porsche* data set post KDE registration.

Figure 72: Stuttgart *17_porsche* result set.



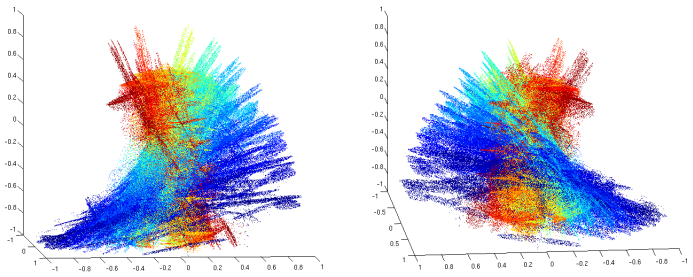
(a) Stuttgart *04_copter* data set after spin image coarse registration.



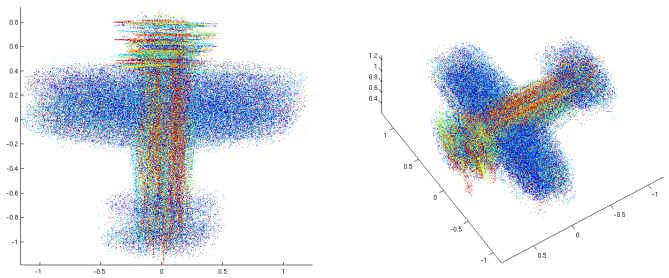
(b) Stuttgart *04_copter* data set post KDE registration. Some minor registration error is still evident on the object main rotor however registration quality visually generally improves (*e.g.* features such as the tail rotor and landing gear.

Figure 73: Stuttgart *04_copter* result set.

Once a coarse alignment has been generated, each data set is provided as input to our KDE multi-view registration algorithm, Scanalyze [210] and the Procrustes method [262]. For the experimental assessment of alignment accuracy we are interested in the registration error produced by each algorithm, for each utilised Stuttgart data set. Global registration results are reported in terms of mean inter-point distance (defined previously in chapter 3) and a summary of results are found at the end of the chapter in Table 10. Sample registration results are visualised in Figures 72, 73, 74 and 75.

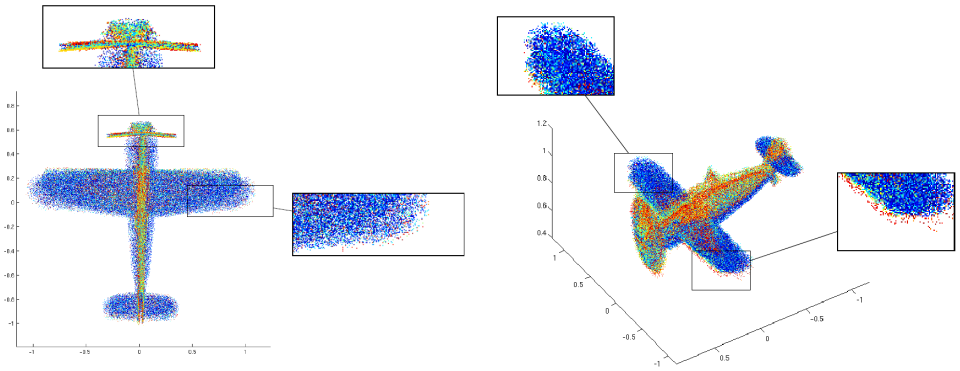


(a) Stuttgart *42_fighter* post depth image reprojection to point clouds. Dataset in pre-coarse alignment configuration hence point sets lack coherent frame of reference. A coarse alignment strategy must be utilised before (any) dense, iterative error minimisation registration approaches can be successfully applied.

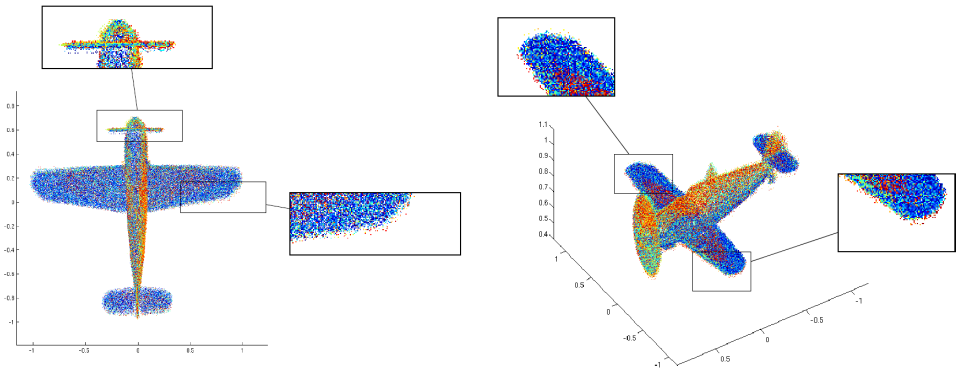


(b) Stuttgart *42_fighter* data set post coarse alignment using the simplistic sparse feature based approach outlined in section 5.2.1.

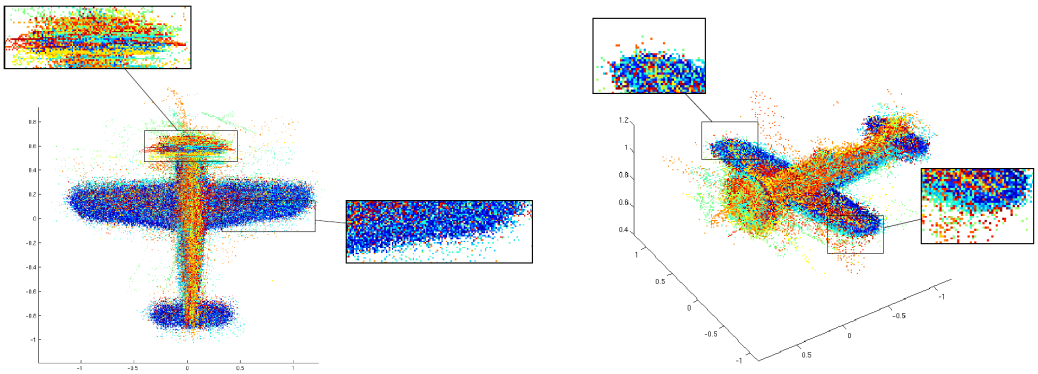
Figure 74: The initial stages of the registration pipeline applied to the Stuttgart *42_fighter* point cloud data set.



(a) Stuttgart *42_fighter* fine registration due to [262].



(b) Stuttgart *42_fighter* fine registration due to the proposed KDE registration technique.



(c) Stuttgart *42_fighter* fine registration due to Scanalyze [210].

Figure 75: Fine registration results. The proposed method shows visually improved registration, sharpening up object areas that are expected to display flush surfaces (*e.g.* tail, wing). The Scanalyze technique [210], although able to scale favourably to large data sets such as this, attempts to optimally align each view with respect to others in a sequential way leading to visualised error propagation. The Procrustes method [262] exhibits good global registration yet some clique formation (*e.g.* propeller area) is still visible.

5.4.3.2 *Stuttgart range images: summary*

The Stuttgart data set experiments allow a number of considerations to be drawn. Superior registration is evident in terms of achievable alignment accuracy across a varied set of object shapes and test cases (see Table 10 for accuracy results). For large data sets, such as those experimented with in this chapter, the Scanalyze technique [210] on occasion fails to reach a visually acceptable global registration and this failure is reinforced by corresponding large quantitative error. When attempting to register large sets of point clouds it can be presupposed that large view-sets are more liable to incur problematic phenomena such as error propagation and loop closure than small view data sets (corroborating other recent related findings *e.g.* [26]). On the other hand, it has also previously been noted [27] that the method underlying Scanalyze tends to scale favourably in terms of computation time when the number of views increase (orders of hundred range images used in this work) and our experimental findings confirm this. In contrast the method of [262] performs a genuine simultaneous multi-view registration and typically generates visually pleasing results yet the cost of true simultaneous registration often results in wall clock runtime similar to the introduced method (*c.f.* Table 10), yet the introduced strategy is in some cases able to produce better global object shape (*e.g.* Figure 75). The main and achieved objective of the current work was to present the ability to perform high quality registration to extremely large view-sets and, although computationally demanding in comparison to the considered alternatives, parallelisation through our introduced SSTF framework (chapter 4) yields a feasible route to applying demanding registration strategies to view-sets containing $100 - 500$ viewpoints whilst maintaining, and often exceeding, the accuracy performance of contemporary alternatives. In conclusion it can be claimed that the proposed registration framework has been demonstrated to be a viable solution for the global registration of large collections (hundreds of views) of dense range images as part of a modern, high-quality 3D object modelling pipeline.

5.4.4 *Stereo video: data sets*

Further real-world experimental work is carried out by capturing depth image data using a stereo video system. Stereo video affords high frame-rate data capture that suits

a wide range of applications from facial animation to high-speed surface deformation analysis. In the work presented here, a commercial DI4D stereo capture system [73] is used to passively obtain rigid object depth information from multiple viewpoints for the task of complete model reconstruction. While model reconstruction from stereo is a well studied topic (see [231] for a comprehensive review), in this work we explore the advantages of reconstructing objects from large sets of depth images captured by this sort of modern stereo video capture equipment. By combining and applying the multi-view point cloud registration techniques introduced previously (chapter 3) and the SSTF framework proposed in chapter 4 we utilise our distributed multi-view point cloud registration pipeline to accommodate data sets containing frame counts on the order of magnitude typically associated with modern stereo video capture equipment (several hundreds or thousands of frames).

The acquired stereo video data provides an opportunity to test the suitability of the introduced registration framework with data sets obtained from an additional and real depth acquisition process. Using the proposed technique with data acquired via depth-from-stereo facilitates the testing of robustness to noise and sensor error distributions typical of stereo data (*e.g.* the effects of systematic geometric and radiometric sensor errors to point set reconstruction [143]).

The DI4D cameras operate at ~ 25 fps and offer high resolution (1 megapixel) depth image sequences. Two monochrome cameras are used to retrieve scene depth information. Image correspondences are calculated, per image pair, by proprietary software [73] in order to provide depth-from-stereo information. A third (colour) camera captures RGB intensity information, aligned to the inferred depth map. We use a standard pinhole camera model to convert each depth image to a 3D point cloud.

By acquiring large numbers of point sets from high frame-rate stereo cameras and combining these with our techniques for high quality, large view-set, point cloud registration we offer evidence that high speed depth-from-stereo systems are a valid route to full and complete 3D model acquisition using only a single sensor. Other recent consumer depth sensor and 3D object acquisition advances [183, 190, 135] make use of alternative high frame-rate sensing techniques (*e.g.* structured light) for depth acquisition. High speed depth-from-stereo combined with multi-view registration techniques, capable of registering many frames, provide a fast and viable *high-resolution* alternative 3D object acquisition pipeline. This is useful where passive, non-invasive depth capture

(*e.g.* from stereo) is a preferred or required system feature. Capture situations, pertinent to many contemporary applications, involve object acquisition where structured light may not be viable. Object acquisition systems making use of an infra-red structured light sensor (*e.g.* a Kinect [183]) are not usable in certain instances (*e.g.* outdoors) due to infra-red information being disrupted (*e.g.* by sunlight). Coupling the passive nature and instantaneous capture of depth-from-stereo (and related techniques making use of large collections of photographs) with appropriate large-scale multi-view registration techniques has previously been shown to extend the settings in which depth inference and multi-view registration can be practically applied (*e.g.* [98],[5],[42]). In this section we further explore the ability of our framework to handle the multi-view registration of large point sets using data provided by real-world, depth-from-stereo sensors.

(a) Bust figurehead object on turntable. A colour camera provides 1040 × 1392 RGB intensity data aligned to monochrome image pairs.

(b) Bust figurehead depth image recovered from a monochrome stereo image pair using the depth-from-stereo algorithm of [73].

Figure 76: Sample colour and depth image frames of the bust figurehead object captured using a 25 fps stereo camera rig [73].

5.4.4.1 *Stereo video: multi-view registration*

A bust figurehead is used as a test object with which to obtain high frame-rate depth images from stereo video (see Figure 76 for an example frame). The bust is placed on a turntable and $\sim$ 10 seconds of stereo video footage is recorded whilst rotating the turntable through one complete revolution. A uniform background colour is provided

to aid object segmentation but no object or scene markers are required. One complete turntable revolution (360° degrees) is recorded resulting in each side of the object being captured in multiple video frames. The camera frame-rate is high enough such that many views (of all sides) of small and medium sized objects can be captured in a relatively short time frame while avoiding movement and capture-speed based problems (*e.g.* motion blur).

In this instance the short capture time provides 220 depth images (individually reprojected to 3D point clouds) covering each side of the bust figurehead. The sensor position is fixed and the bust remains in a fixed position on the turntable. Object views from above and below (that would capture the crown and base of the bust) are therefore omitted. Image correspondences and resulting depth maps are provided for each stereo image pair by proprietary DI4D software [73] and after extracting point clouds from the depth images, point sets are again coarsely aligned using the method described in section 5.2.2. The full data set offers dense depth maps containing $\sim$ 1.5 million points per depth image and therefore the entire data set consists of $\sim$ 300 million points (pre object segmentation). To aid processing, down-sampling is again (*c.f.* chapter 3, section 3.5.2) performed on each viewpoint. Viewpoint point clouds are down-sampled uniformly to $\sim 0.2\%$ to enable feasible experimentation with all considered registration methods and implementations. Our implementation of [262] is via serial work station, typically unable to accommodate view-set problem instances with point sets containing millions of points[4]. In summary a multi-view registration comparison is performed utilising recent methods and point set magnitudes that can be considered challenging in terms of both *point density* and *view-set size* provided by a modern depth sensor.

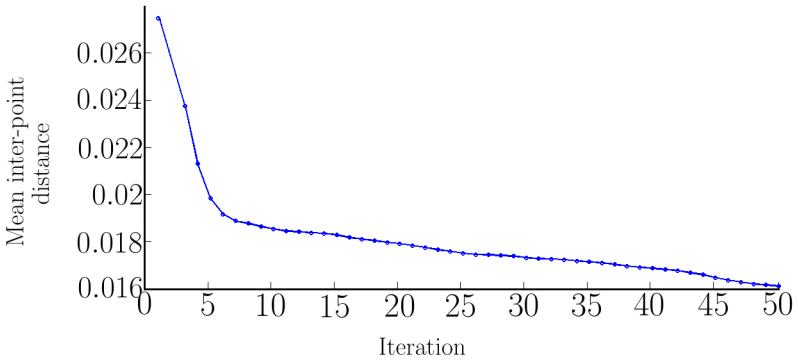
By using the method described in section 5.2.2 coarse alignment is achieved by again assuming a constant turntable rotation rate in the z-axis during capture. Additive initial rotations of $\frac{360^\circ}{220} \approx 1.636^\circ$ are consecutively applied to each viewpoint. These rotations, in addition to some manual translation, again provide the coarse view alignment in a global frame of reference (views of the resulting coarse alignment are found in Figures 79a and 79d). This alignment is then provided as input to both the proposed multi-view registration algorithm and that of Toldo et al. [262].

4 This is due to the local serial implementation of the Procrustes method and available hardware rather than a claim about theoretical properties of the method.

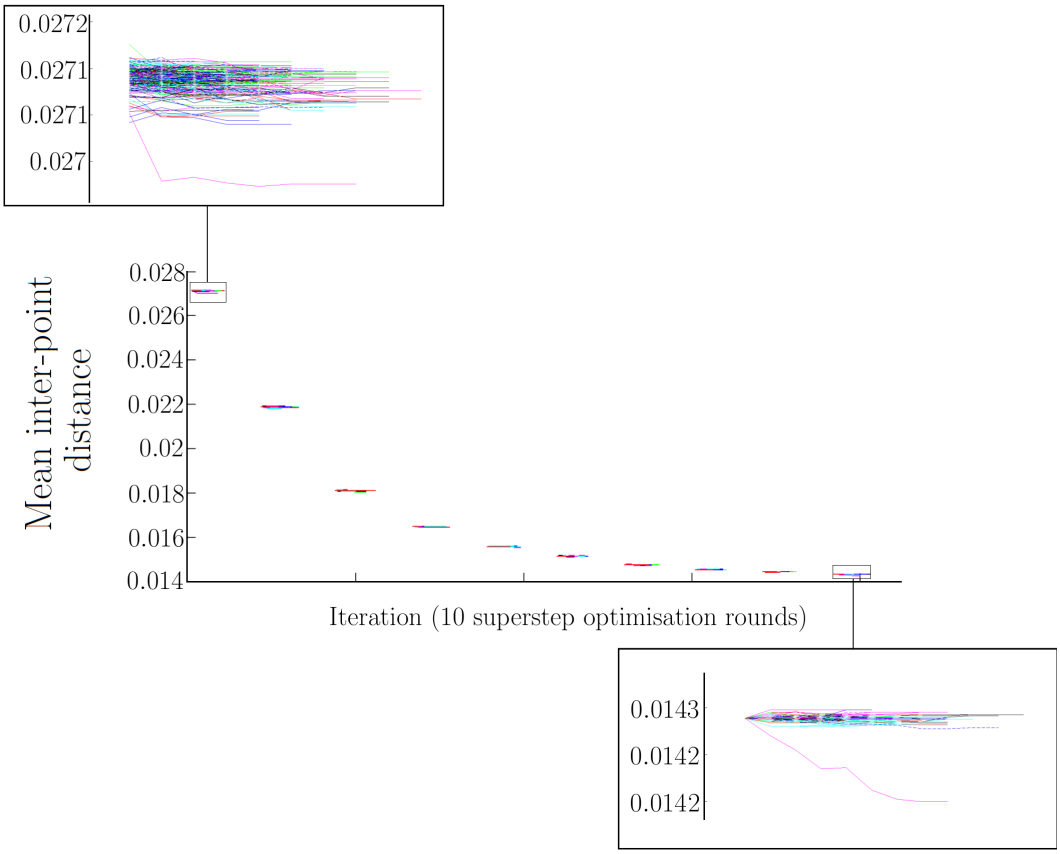
5.4.4.2 *Stereo video: convergence and stopping behaviour*

We show mean inter-point distance (μ_{ipd}, see section 3.5.1.1 for explanation of why this is an appropriate measure) evolution during registration for the bust figurehead data set in Figure 77. In both subfigures (pertaining to each method) transform update iterations are reported on the horizontal axis (this is not informative of computation time but does provide insight into convergence behaviour). After ten iterations the introduced multi-view registration method is shown to be near convergence in terms of μ_{ipd} whilst the method of Toldo et al. [262] can be seen to be still slowly reducing the μ_{ipd} error after 50 iterations (as noted this is not informative of computational cost). The error metric in the case of the Toldo et al. result can be seen to begin to slowly (yet not completely) converge, however allowing the algorithm to proceed further (> 50 iterations) results in two "clique" like point cloud subsets enjoying increasingly tight inter-clique registration and drifting closer together while failing to capture and reproduce true global object shape, yielding visually unsatisfying results (*c.f.* Figure 79e).

For the introduced registration method, Figure 77b shows μ_{ipd} evaluated for *each* scan optimisation in parallel (where μ_{ipd} is defined over the points belonging to the scan being optimised) at *every* transform space optimisation iteration. As discussed in chapter 3 our transform optimisation is performed using Quasi-Newton line search, refer to chapter 3, sections 3.5.1.1 and 3.4 for the μ_{ipd} quality metric and optimisation technique details.



(a) Evolution of the μ_{ipd} registration metric using the Toldo et al. [262] multi-view registration method. See text for discussion.



(b) Iterative error metric μ_{ipd} evolution using the proposed multi-view registration approach. Large improvements are achieved in the early supersteps (individual, yet parallelised, transform space optimisation) and the alignment quality is refined as surface re-estimation proceeds with each superstep. Enlarged insets show latter supersteps, in addition to collectively displaying low absolute μ_{ipd} values, also exhibit less fluctuation and lower inter-scan μ_{ipd} variance.

Figure 77: Value of the μ_{ipd} (mean inter-point distance) registration quality metric during iterative multi-view registration of the bust figurehead data set performed via the Toldo et al. algorithm (77a) and the proposed method (77b).

For the examined stereo depth data set, the largest error reduction in both techniques can be attributed to early iterations (similar behaviour was observed for both (1) the small synthetic data sets experimented with in chapter 3 and (2) the large view-set synthetic data in this chapter, section 5.4.2.1. The introduced registration process is limited here to ten supersteps as the method was previously shown to produce acceptable registration results with this size of iteration cap for synthetic data (see section 5.4.2.1). We concede that each iteration of the proposed method is computationally more expensive than an iteration of the Toldo et al. [262] algorithm. A superstep iteration of the introduced method constitutes parallelised local transform space search, effectively performing true simultaneous view registration.

For the figurehead data set (220 views) the introduced method performs, as noted, a pose optimisation for each point cloud in parallel during each optimisation round (*superstep*). The enlarged insets in Figure 77b show a parallelised superstep round with one viewpoint transform space optimisation represented per colour. Each superstep round contains a maximum of 50 optimisation steps per viewpoint and horizontal axis separation between rounds is introduced for expository purpose to highlight superstep completions. Separations therefore indicate where each round of viewpoint transform space optimisation ends and where new surface approximations are estimated using the updated point cloud positions.

It can be observed that the latter supersteps, in addition to collectively displaying low absolute μ_{ipd} values, also exhibit decelerating improvement and lower inter-scan μ_{ipd} variance (see Figure 77b zoom insets), providing further evidence of procedure convergence in only 10 iterations.

Interestingly, as each scan moves in parallel, optimising its position in relation to the inferred object surface, it can be seen that the obtained μ_{ipd} values (and similarly RMS metrics, not shown) do not provide strictly decreasing functions of μ_{ipd} error in every view due to simple point pair distances not being directly minimised in the objective function. It is thought that by not directly minimising a simple point-pair distance metric the method may be able to make use of information, potentially collected from many view-points, to make *globally better* transform space updates. The μ_{ipd} metric does however decrease as surface estimates update after each *superstep* completion and therefore it can be concluded that even if individual scan positions are updated to locally

sub-optimal positions (in terms of point pair distance) this can be globally beneficial in terms of overall object surface shape and multi-view registration error distribution.

Final error metric values for coarse alignment and converged registration are displayed in Figure 78 and the respective point set configurations that generate these are displayed in Figure 79.

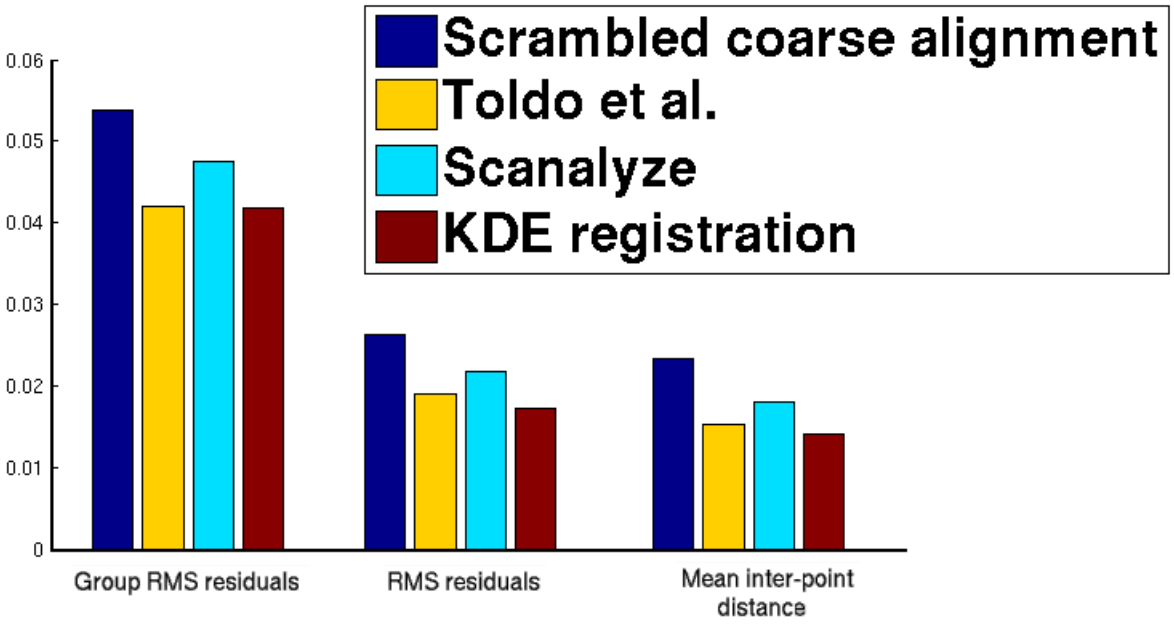


Figure 78: Error measures used to evaluate both the initial coarsely aligned bust view configuration and the converged registered view-sets, generated by the evaluated registration methods. The bust illustrates a data set where the quantitative error metric difference between registered view-sets is small yet difference in visual appearance is pronounced (*c.f.* Figure 79).

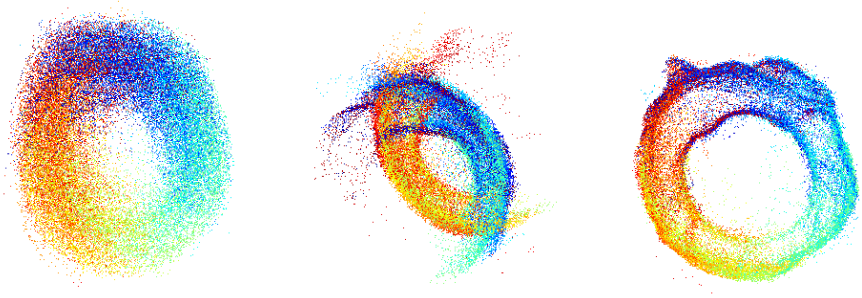
For the bust data set, the difference in visual appearance between resulting registered point sets is pronounced yet the related differences in quantitative error metrics are found to be *not* statistically significant. As in chapter 3, repeated trials, seeded by random coarse alignment, might be utilised to reveal a valid (small) effect size but the point we highlight here is that visually disparate outcomes can yield quantitatively similar results when employing standard error metrics commonly used to promote the capabilities of point set registration algorithms. Visual inspection in most cases offers a valuable, valid additional tool when assessing registration performance. This suggests an additional direction for future work involving investigating or employing (*e.g.* [245])

intelligent multi-view registration error metrics shown to make the measures less susceptible to this phenomenon.

(a) Coarsely aligned bust figurehead point clouds: front-to-parallel view.

(b) Registration result of Toldo et al. [262] using 50 iterations.

(c) Registration result using the introduced method, using 10 iterations (supersteps).



(d) Coarsely aligned bust figurehead point clouds: view orthogonal to turntable planar surface.

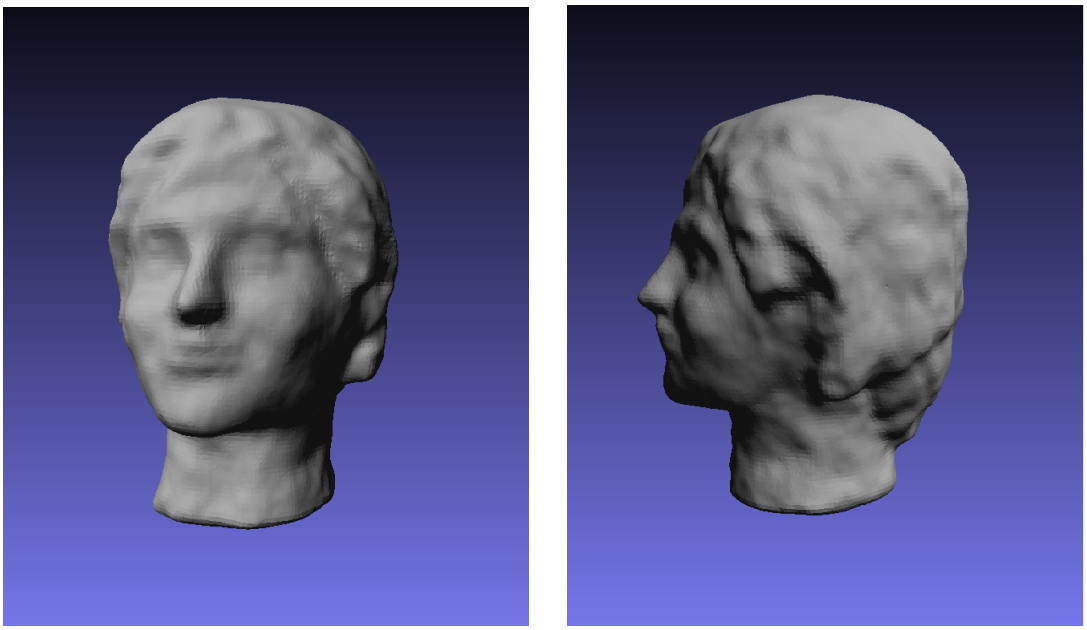
(e) Resulting registration of Toldo et al. [262] using 50 iterations.

(f) Registration result using the introduced method, using 10 iterations (supersteps).

Figure 79: Views of the coarse alignment configuration are shown in 79a, 79d (see text for coarse alignment details). Coarse alignments are provided as input to two registration algorithms, views of registration results are provided in 79b, 79c, 79e and 79f. The top row displays a profile view whilst the lower row displays a position underneath the bust with view directed up through the central object z-axis, exposing registration results (note lack of depth data pertaining to object base and crown-of-head).

For illustrative purposes, and to complete the reconstruction pipeline, we provide a watertight object model derived from our set of 220 depth images, captured with our stereo video camera rig [73]. The derived point cloud view-set, post simultaneous multi-view registration using the introduced method, is provided as input for surface reconstruction.

For the task of surface reconstruction we again make use of the implicit reconstruction technique, Poisson surfacing [145]. It can observed that the Poisson surfacing result (Figure 80) that is obtained by applying surface reconstruction to the registered point set provides a geometrically recognisable model of the original object (*c.f.* 2D RGB intensity data input, Figure 76a). The reconstructed surface here uses an amalgamated point set, consisting of all point samples from 220 depth images, as input. The reconstruction is visually similar to the original object on account of the robust registration strategy employed however *point set integration* is a related area of work that aims to integrate multiple 3D scans intelligently, post-registration, in order to improve reconstruction quality and surface integration (*e.g.* the multi-scale saliency based approach of [244]). Further exploration of intelligent *point set integration* for large view-sets, in combination with feasible registration techniques, provides a further promising area of further work.



(a) (b)

Figure 80: Bust data set containing 220 registered views using the introduced multi-view registration technique. Aligned point sets are amalgamated and a Poisson surfacing technique applied to produce a watertight object model (*c.f.* 2D RGB intensity data input, Figure 76a).

5.4.4.3 *Stereo video: summary*

In conclusion, our depth-from-stereo sensor and bust figurehead data set exploration allow further conclusions to be drawn. In a similar fashion to the spray bottle object experimented with previously (chapter 3), we conjecture that our approach is able here to produce the most reasonable visual result by optimising scan positions in relation to our global surface estimate and soft correspondence strategy. The introduced registration strategy enforces correct global shape consistently when many scans (containing relatively simple geometric structure in this case) overlap and contribute to object shape. This scenario is likely when many object views are available, potentially providing redundant depth information. By utilising (and iteratively refining) inferences regarding global object shape we provide information to influence optimal local registration of each point cloud while implicitly taking into account global structure and cohesion. By avoiding the explicit minimisation of hard local point pair correspondence distances we attempt to ensure that individual alignment does not drive view positions to locally optimal yet globally poor poses that are detrimental to global shape coherence. Using the introduced strategy of inferring global object shape via density estimation proves expensive for large data sets yet methods that lack a concept of a global object shape may exhibit gross failure modes. We note that these experimental results are due in part to favourable coarse alignment seed positions allowing the introduced registration technique to produce reasonable initial inferences regarding object surface (see *e.g.* Figure 79d).

5.5 DATA SETS SUMMARY AND DISCUSSION

Here we quantitatively summarise the tests and comparisons performed in this chapter involving the proposed solution, the global registration technique originally introduced by Pulli [210] and the Procrustes approach suggested in [262]. Algorithm implementations make use of varying systems with the proposed technique (implemented in Matlab) utilising our local ECDF distributed resource [82], the method of Pulli (implemented in C++) run on a PC AMD Athlon 64bit Dual Core (2×3.00 Ghz) with 3.00 GB of RAM and the method of Toldo et al. [262] (implemented in Matlab) utilising the same local system.

Registration results are collated and summary tables (Tables 10, 11) list accuracy and computation time for all data sets made use of in this chapter. Mean inter-point distance (μ_{ipd}) values are collected for coarse pre-registration configurations and corresponding values for converged registration configurations, for each algorithm experimented with. For experimental assessment of the registration accuracy we evaluate the registration error produced by each implemented algorithm for all utilised datasets. The first column of Table 10 lists the data source while columns two and three provide information about the number of views of each considered dataset and the average scan densities respectively. Column four reports the mean inter-point distance among the views associated to their initial coarse alignment condition. Global registration results are then reported in terms of mean inter-point distance for each of the considered fine registration methods. Note that the method of [210] failed to return a registration for the data set *04_copter* due to memory exhaustion caused by the attempted computation of global registration, reaching the upper limit of available GPU memory (as confirmed by a system process monitor), and therefore generated a runtime exception. Nevertheless, the Pulli [210] registration on smaller data sets provides indication of typical performance and we would not expect it to function substantially differently, in terms of registration quality, if more memory were afforded to handle a dataset of size similar to that of *04_copter*.

Table 10: Global registration error metrics for large view-sets.

Data set	Number of viewpoints	Mean # points per view	Coarse alignment μ_{ipd}	KDE registration μ_{ipd}	Scanalyze [210] μ_{ipd}	Toldo et al [262] μ_{ipd}
Real Tridecahedron	522	2720	2.372567	**2.006429**	2.05299397	2.212345
Synthetic Tridecahedron (± 0.0 , $\sigma = 0.00$)	250	5000	**0.003444**	0.003526	0.01199586	0.004753
Synthetic Tridecahedron (± 0.175 , $\sigma = 0.00$)	250	5000	0.012466	**0.003659**	0.01278884	0.006491
Synthetic Tridecahedron (± 0.350 , $\sigma = 0.00$)	250	5000	0.015029	**0.004723**	0.01492897	0.008447
Synthetic Tridecahedron (± 0.500 , $\sigma = 0.00$)	250	5000	0.016256	**0.004524**	0.01667278	0.008874
Synthetic Tridecahedron (± 0.0 , $\sigma = 0.01$)	250	5000	**0.006702**	0.006741	0.01232697	0.007023
Synthetic Tridecahedron (± 0.175 , $\sigma = 0.01$)	250	5000	0.013272	**0.004620**	0.01276762	0.008106
Synthetic Tridecahedron (± 0.350 , $\sigma = 0.01$)	250	5000	0.016227	**0.006702**	0.01577113	0.009482
Synthetic Tridecahedron (± 0.500 , $\sigma = 0.01$)	250	5000	0.017941	**0.008436**	0.01773417	0.010261
Synthetic Tridecahedron (± 0.0 , $\sigma = 0.02$)	250	5000	**0.008296**	0.008341	0.01295305	0.008499
Synthetic Tridecahedron (± 0.175 , $\sigma = 0.02$)	250	5000	0.013354	**0.008288**	0.01310442	0.009298
Synthetic Tridecahedron (± 0.350 , $\sigma = 0.02$)	250	5000	0.016936	**0.008318**	0.01689198	0.010561
Synthetic Tridecahedron (± 0.500 , $\sigma = 0.02$)	250	5000	0.018571	**0.008840**	0.01773311	0.011160
Synthetic Tridecahedron (± 0.0 , $\sigma = 0.04$)	250	5000	**0.010239**	0.010276	0.01438224	0.010303
Synthetic Tridecahedron (± 0.175 , $\sigma = 0.04$)	250	5000	0.014071	**0.010210**	0.01344889	0.012845
Synthetic Tridecahedron (± 0.350 , $\sigma = 0.04$)	250	5000	0.017319	**0.010218**	0.01524040	0.011789
Synthetic Tridecahedron (± 0.500 , $\sigma = 0.04$)	250	5000	0.019795	**0.010283**	0.01881461	0.010897
Synthetic spheres (large point cloud viewpoints)	250	50000	0.012606	**0.004881**	—	—
42_fighter	258	7953	0.007701	**0.002922**	0.00646944	0.003205
17_porsche	258	16094	0.008564	**0.005943**	0.00781996	0.009030
04_copter	258	19846	0.007624	0.004669	—	**0.004193**
Bust figurehead	220	2209	0.024602	**0.012239**	0.01824514	0.016057

Table 10 suggests that our registration framework is able to provide reduced fitting error compared to both Scanalyze [210] and Procrustes [262] algorithms for large view-set data. The KDE registration method affords the lowest mean inter-point distance error for all datasets experimented with (apart from *04_copter*) and importantly is able to follow linear convergence rates (see *e.g.* Figures 60, 55) towards such final viewpoint configurations. While for some datasets experimented with (*e.g. Synthetic Tridecahedron* ± 0.175,$\sigma = 0.01$ and *17_porsche*) all three techniques converge to a visually acceptable minimum, in other cases (*e.g. Bust figurehead*, *42_fighter*) the methods of Pulli [210] and Toldo et al. [262] fail to reach a visually agreeable configuration and in many (but not every) cases exhibit a corresponding high quantitative registration error. As noted as view count increases, large view-set data seems generally more liable to incur problematic phenomena such as error propagation and loop closure.

In Table 11 we list computation times of the various registration strategies measured in minutes. The KDE registration wall-clock and idealised run times are as described previously (section 4.5.1.1). The method of [210] is clearly faster than the proposed method in reaching its error minimum, especially for larger datasets. While the Pulli optimisation [210] uses a sequence of ICP applications, which are very fast to compute, our KDE registration considers the alignment globally, causing the computational burden of the approach to increase linearly with the number of scans to be processed (partially mitigated here by our distributed framework and implementation). A possible future direction, in this area of computational expense mitigation, would involve exploiting an initial (fast) registration produced by *e.g.* [210] to bring an alignment closer to the optimal registration and seed the introduced method with this such that the proposed approach can converge faster. A similar strategy was explored recently (in [27]) however no bold conclusions could be drawn from the early stage of their experimentation.

Computational behaviour is seen to fluctuate in relation to dataset size for both global registration methods investigated. In some cases (*Synthetic Tridecahedron* ± 0.175,$\sigma = 0.01$ and *42_fighter*) our approach takes moderately longer than that of Toldo et al. [262], in terms of wall-clock time however the idealised time (see section 4.5.1.1) is lower indicating that queueing effects should be taken into consideration when assessing the feasibility of utilising shared distributed resources. In other cases (such as *Bust figurehead*) our method is able to afford an improvement in terms of the computation

time required. In short, it can be concluded that the computational performance of our approach is dataset-dependent, however, the ECDF cluster based implementation affords an effective route to mitigate the computational burden of larger datasets.

Further to this, it is noted that runtime comparison between methods is not directly meaningful in terms of computational analyses due to the studied algorithms differing in (1) implementation language and (2) computational platform utilised. We include run times for informational purpose and note that although the introduced method is an order of magnitude slower than the (C++ implementation) of Pulli's optimisation technique [210] we are often able to produce higher quality registration results (with our, comparatively expensive, global method). Our distributed approach allows for run times that remain feasible (*e.g.* broadly similar to those of the Toldo et al. [262] algorithm) even when applying computationally demanding registration techniques to very large data sets, yielding high quality results.

Table 11: Multi-view registration computation timings (minutes)

	KDE Registration Timing (minutes)		Scanalyze [210] Timing (minutes)	Toldo et al. [262] Timing (minutes)
Data set	Ideal time	Wall-clock time	Wall-clock time	Wall-clock time
Real Tridecahedron	97.14	473.32	8.05	299.42
Synthetic Tridecahedron (± 0.0 , $\sigma = 0.00$)	35.43	49.92	6.92	475.60
Synthetic Tridecahedron (± 0.175 , $\sigma = 0.00$)	117.25	421.30	6.11	478.99
Synthetic Tridecahedron (± 0.350 , $\sigma = 0.00$)	146.12	427.87	6.36	481.41
Synthetic Tridecahedron (± 0.500 , $\sigma = 0.00$)	119.46	412.22	6.00	470.14
Synthetic Tridecahedron (± 0.0 , $\sigma = 0.01$)	97.92	395.59	6.20	481.49
Synthetic Tridecahedron (± 0.175 , $\sigma = 0.01$)	92.88	418.12	6.67	466.18
Synthetic Tridecahedron (± 0.350 , $\sigma = 0.01$)	95.97	406.87	6.86	446.63
Synthetic Tridecahedron (± 0.500 , $\sigma = 0.01$)	111.79	423.58	6.74	446.64
Synthetic Tridecahedron (± 0.0 , $\sigma = 0.02$)	38.29	60.38	6.05	479.44
Synthetic Tridecahedron (± 0.175 , $\sigma = 0.02$)	84.84	618.60	6.36	442.43
Synthetic Tridecahedron (± 0.350 , $\sigma = 0.02$)	89.99	593.54	6.56	450.50
Synthetic Tridecahedron (± 0.500 , $\sigma = 0.02$)	91.27	598.78	6.74	450.10
Synthetic Tridecahedron (± 0.0 , $\sigma = 0.04$)	41.12	68.92	6.92	487.83
Synthetic Tridecahedron (± 0.175 , $\sigma = 0.04$)	89.01	205.80	7.49	450.14
Synthetic Tridecahedron (± 0.350 , $\sigma = 0.04$)	91.13	201.69	7.30	446.41
Synthetic Tridecahedron (± 0.500 , $\sigma = 0.04$)	89.18	193.70	6.35	437.02
Synthetic spheres (large point cloud viewpoints)	1281.31	1729.52	—	—
42_fighter	74.56	143.59	4.68	183.45
17_porsche	104.85	180.06	11.89	222.96
04_copter	137.54	271.85	—	1051.91
Bust figurehead	66.22	344.39	2.89	148.35
Implementation language	Distributed Matlab		C++ and CUDA	Matlab
Max # of cores utilised	70		2	2

In conclusion, as confirmed by the results collected (Tables 10, 11), heuristic registration methods can be faster yet their convergence is not guaranteed. On the other hand our KDE registration method remains slower than reference heuristic-based methods, with a runtime performance gap that tends to increase for larger datasets where many point samples must be evaluated. Despite our effort and achievements in finding expedients to reduce the computational burden of our method, further computational optimisations are still possible (*e.g.* parallel computed correspondence finding for kernel estimation) which are left to further works. What we propose and test here is the performance evaluation of a distributed scheme capable of undertaking large view-set registrations and producing high quality results in feasible time frames. This results in mitigating the high computational cost associated with large view count global registration and allows for observations that prove useful for the design of feasible large-scale registration strategies. In this way we aim to bring the scan configurations to an optimal solution while mitigating the related optimisation engine workload.

Finally, we expect that computational performance of the introduced method can be sensibly improved by surpassing some limitations of our current implementation. Possible routes to this end include following previously successful strategies involving parallelising work at a finer granularity (*e.g.* parallelising expensive point set nearest neighbour search via GPGPU [253], [211] rather than at the coarser distribution level of transform space search). By performing this correspondence selection at each iteration via computation on GPU hardware we could expect significant runtime improvement. Additionally, reimplementation in a compiled language (*e.g.* C++), whilst retaining our SSTF framework would also result in absolute runtime speed up.

The main and achieved objective of the chapter was to present the ability to perform high quality registration to (what are currently) large view-sets and, although computationally demanding in comparison to the compared alternatives, parallelisation through our introduced SSTF framework (chapter 4) yields a feasible route to applying demanding registration strategies to high order of magnitude view-sets whilst fully maintaining documented registration accuracy benefits. It is claimed that the presented combination of registration strategy and distributed task farming framework is an attractive option for performing global registration of large collections of dense point clouds as part of a modern, high-quality 3D object modelling pipeline.

Part VI

DISCUSSION

6

DISCUSSION

6.1 SUMMARY OF THE THESIS

This thesis has explored the challenge of performing point set registration where many sets of 3D points must be considered. In this work point sets typically represent spatial measurement of physical environments or objects from varying viewpoints and thus require global alignment into a common frame of reference to provide useful input for *e.g.* the subsequent stages in a model acquisition and reconstruction pipeline. Multi-view point set registration is challenging primarily due to the large amount of variability found in complex objects, environments and additionally the large amount of data that must be treated. The sources of variability include, but are not limited to, sensor capture rates, object shape and surface properties, sensor (and object) trajectories and scene illumination. In this thesis we focussed on developing principled models that allow us to incorporate knowledge about local object surface shape into solving the point set registration task and investigated feasible routes to applying these methods to large point set data.

As highlighted in the introduction, modelling pipelines that produce accurate 3D models of complex physical objects and environments can be utilised in many useful application areas. This observation has motivated the large body of work that exists on automated 3D modelling, treated in our literature review. The approach to model reconstruction typically involves first acquiring partial 3D point sets of an object from each captured viewpoint, aligning these partial sets together and fusing all partial views to obtain a full, compact and potentially watertight object representation. Aligning the acquired depth samples can be regarded as the most limiting step of the 3D modelling pipeline and this problem is compounded in difficulty as the number of viewpoints

increases due to both the involved optimisation principles and computational considerations.

In real-world examples, this alignment problem proves challenging due to the large amount of variability one sees in objects found in the natural world. Factors include, but are not limited to, object pose, appearance and shape, camera pose and scene illumination. When depth information capture can be constrained to only consider *e.g.* highly accurate sensors, reliable depth inference, controlled object presentation, reliable and uniform viewpoint sampling devoid of occlusion, then simple chained pairwise view registration might be utilised to quickly and frugally provide full model reconstruction. If view chains become long, the worries of error accumulation and propagation remain however this approach may prove satisfactory under the highlighted optimal conditions. Such conditions and the resulting samples are however not always representative of practical inputs that a real-world registration algorithm might be expected to deal with. In realistic conditions, depth samples often exhibit noise, occlusion, clutter and realistic viewpoint sampling may present larger variance in sample positions and orientations of objects of interest.

An important premise of this thesis alleged that large view-sets and an abundance of depth data are beneficial in terms of model completeness and accuracy and can be used advantageously when tasked with modelling object surfaces and shape. In this work we hypothesised that view alignment accuracy and robustness can be improved over sequential registration approaches by employing simultaneous registration to inherently take advantage of information contained in many viewpoints and distribute misalignment errors between overlapping views. This in turn allows object surfaces to be robustly and reliably estimated from coarsely misaligned views in a data-driven fashion in order to inform and drive the view registration process. We formalised this hypothesis with the claim:

By registering partial views simultaneously to a robust surface estimate, it is possible to improve registration accuracy over sequential approaches by distributing errors evenly between overlapping viewpoints. Object surfaces can be robustly estimated from coarsely misaligned partial views using density estimation techniques and such estimates can be utilised to reliably guide simultaneous point cloud registration. This approach exhibits an inherent ability to handle data from many viewpoints simultaneously and improves

registration and reconstruction accuracy over existing techniques by exhibiting robustness to initial coarse misalignment of view-sets.

Our work has defended this thesis by presenting the following original contributions:

(1) We propose new multi-view registration techniques that leverage an abundance of viewpoint information for the registration task. Chapter 3 focussed on developing statistical density models that allowed us to robustly reason about local surface and shape. In particular, by proposing new methods to simultaneously register multi-view point cloud data this work has sought to further knowledge and understanding of point set registration challenges and problems that occur as the number of viewpoints increase. In this chapter we provided quantitative evidence, by way of statistical error measures, to illustrate registration quality and indicate improvement over both historically popular and recently proposed multi-view registration approaches (see section 6.1.1).

(2) We introduce a novel task-farming framework that facilitate accurate simultaneous registration of large sets of point clouds in a global coordinate frame. Both the computational speed of density estimation and quality of resulting models typically depend on the number of data samples available. Intrinsic properties of non-parametric density estimation dictate that estimation quality improves as the number of available samples increases however estimation often also becomes more expensive. This non-parametric estimation property essentially dictates that the cost of building models will increase as the number of available samples to be utilised increases. In Chapter 4 we introduced a task distribution strategy offering effective methodology for solving computationally expensive problems and contribute quantitative evidence of the obtainable *predictable* speed improvement (see section 6.1.2). The framework is generically applicable however in this thesis it enabled investigation of the claim that for *many viewpoints*, a data-driven simultaneous approach is able to improve the registration process.

(3) Finally Chapter 5 contributed an investigation into performing the multi-view registration task using extremely large view-sets. In this chapter we fulfilled pragmatic goals of the thesis and provided confirmation that we contribute a registration methodology

suitable for real-world use. By investigating data consisting of many views from varying viewpoints, this chapter corroborated the hypothesised level of accuracy and robustness that the ability to successfully perform large-scale registration in a simultaneous, data-driven fashion is able to provide. By providing qualitative and quantitative evidence evaluating both registration accuracy and robustness to noise and initial misalignment for view-sets typically larger than those considered in many previous works we provide evidence in support of our initial claim (section 6.1.3). In this chapter we now summarise the outlined thesis contributions, consider conclusions that can be drawn from our findings and highlight potential related areas for future work.

6.1.1 *Kernel Density Estimation for point cloud registration*

In Chapter 3 we proposed the use of density estimation theory to construct surface estimates from point samples and, utilising these estimates, construct novel measures of point set alignment quality. Optimising these measures of alignment quality led to a novel registration process that allows multi-view point set registration to be performed simultaneously for all viewpoints without requiring explicit view-order information or (the typical) point pair correspondence search during the optimisation of viewpoint spatial positions. By avoiding explicit point pair matching we remove one of the computationally expensive parts of a traditionally registration process and by allowing all viewpoints to move in the transform space simultaneously we show typically improved multi-view registration accuracy over sequential alignment approaches.

Soft point correspondence approaches have previously been shown to perform favourably when tasked with handling sample noise and outliers and our experimental work in this chapter additionally supports the stance that *soft correspondence strategies possess the ability to favourably tackle registration problems containing measurement noise and outliers*. Additionally by attempting to solve simultaneously for the global registration of all viewpoints we show that an interdependence between overlapping views can be harnessed to implicitly introduce additional constraints on viewpoint spatial configurations, typically driving the global registration error down.

Synthetic point data sets were utilised to perform experimental validation and illustrate registration reliability, algorithm correctness and robustness to data containing noise. Synthetic data experiments illustrate that our registration process is able to con-

verge to globally optimal viewpoint configurations consistent with known ground truth configurations, even when making use of only rough estimations of the true underlying generating object surfaces. Further to this, active depth acquisition sensor data provided real-world measurements that are inherently corrupted by physical sensor noise. In addition to providing more challenging registration test environments, this allowed for an investigation of how best to select kernel bandwidths for multi-dimensional point cloud density estimation. While many successful data-driven bandwidth selection techniques have been proposed we find that the popular yet expensive task of estimating density derivatives for optimal bandwidth choice can be avoided by utilising simple selection strategies *i.e.* defining bandwidth in relation to sampling density. The relation between appropriate bandwidth selection and resulting estimate smoothness is well understood and important. In practice we find that reliable bandwidth selection manifests as an intrinsic robustness to typical registration challenges involving sampling noise, viewpoint coarse misalignment seeding and "view-clique" convergence problems. Without a need to construct complex sensor noise models, we are able to demonstrate successful registration results using depth data from viewpoints that are seeded with only coarsely defined alignment. By evaluating results across varied data sets under both visual inspection and common statistical registration error measures, quantitative and reproducible evidence in support of our claims regarding resulting accuracy and robustness is provided.

More generally, our registration experiments with real-world data further the argument that registration objective functions, founded on non-parametric principles, provide a better alternative to traditional hard point pair correspondence based metrics for the task of multi-view registration, particularly in cases where robustness to sensor noise is required. Additionally, our experimentation supports the claim that the introduced methodology may prove applicable and useful in pipelines utilising real-world depth data where registration forms a vital component. Finally, Chapter 3 illustrated that our approach can be used in conjunction with common surface reconstruction methods (*e.g.* [145]) to produce representative model surfaces giving further weight to the specific claim that our registration framework may be integrated to form part of an object acquisition and model reconstruction pipeline.

In summary, Chapter 3 presented contributions towards solving a common step in the model acquisition and reconstruction pipeline for complex environments and objects represented by depth measurements from multiple viewpoints.

6.1.2 *Semi-Synchronised Task Farming*

Chapter 4 of this thesis proposed a model for executing intensive large-scale computational problems that contain a mixture of independent and shared (non-independent) problem components that must be integrated to reach a global solution. We name this framework Semi-Synchronised Task Farming (SSTF) due to the affinity with a standard task farming model. The steps of the SSTF framework iterate between distributed independent task computation and information collation steps. After a round of distributed, independent task computation, results are collated and communicated to influence the initialisation and parameterisation of a following round of independent task distribution. This iterative procedure of task distribution and result collation leads to a framework capable of reaching global solutions for problems that can be formulated under the model. In this chapter the attributes and capability of our distributed model were explored by instantiating the framework using local HPC resources.

An additional contribution of Chapter 4 involved the introduction of a related computation time prediction model used to infer total solve time for problems formulated under our SSTF framework. We validated this model using simulated and experimental results and find it to be sufficiently accurate and reliable thus providing a simple tool that could be used when estimating time requirements of computationally expensive algorithms containing distributed elements. By providing an informed model of how execution time depends on input under our framework we provide a useful predictor for distributed problem instances. We concede that producing such a model automatically is not a tractable problem and our timing model is deduced through empirical means, finding key variables that influence computation time. We fit experimental performance to custom functions yet experimentally demonstrate a high degree of predictive accuracy. This timing model proved a useful predictive tool when considering the benefit of implementing algorithms under our distributed framework. It may also lead to accurate computation prediction under additional distributed frameworks that share task parallelisation and result collation components.

The contribution of the SSTF framework allows developers to concentrate on domain specific aspects of computationally expensive problems [178]. Experimental results additionally confirmed that processing data using algorithms formulated under our distributed framework were able to obtain significant time saving over single node computation when deployed on suitable hardware (as might be expected). While the introduced distributed framework is widely applicable, it has hitherto been utilised to aid the formulation of several disparate, yet comparably computationally demanding, contemporary computer vision problems in practice [178].

In summary the work carried out in Chapter 4 introduced a task distribution strategy offering effective methodology for solving computationally expensive problems and resulted in vast wall-clock time savings over analogous serial problem implementations. Our contributions in this chapter consisted of a task distribution strategy for formulating demanding problems that require a level of communication between subtasks and the related, computation-time prediction model.

6.1.3 *Distributed large scale point set registration*

In Chapter 5 we implemented our multi-view registration strategy (introduced in Chapter 3), previously shown to produce accurate and robust multi-view registration results, to view-set collections an order of magnitude larger than those traditionally treated when undertaking multi-view simultaneous alignment. By implementing our strategy using the SSTF framework introduced in Chapter 4, we contribute methodology capable of feasible large view-set registration while still making use of computationally demanding registration framework elements capable of producing high quality results *i.e.* utilising soft point correspondences for alignment evaluation and optimisation involving simultaneous view registration strategies.

By performing experimentation using both simulated synthetic data and data collected from commodity high sampling-rate depth sensors (*e.g.* Microsoft Kinect, video based stereo-camera rigs) we tested the suitability of our novel multi-view registration algorithms for use with high sample-rate data. This offers evidence in support of our claim that *combining demanding simultaneous global optimisation and soft correspondence registration strategies with distributed task farming is an attractive option for performing view registration on large collections of point clouds*. By distributing the

work load of the algorithm, we are able to handle lack of view ordering information, robustness to measurement noise and point outliers and solve for the optimal spatial positioning of viewpoints simultaneously for large-scale point cloud problem instances. This reinforces the point that a distributed instantiation of our framework is a valid step in a modern 3D object modelling pipeline utilising *e.g.* contemporary high speed depth sensors for data acquisition and measurement. A number of quantitative experiments illustrate performance improvements over both historically popular and recent, independently proposed works on a number of benchmark datasets.

6.2 DISCUSSION

The surface approximation models that this thesis proposed, the associated alignment-quality metrics, view optimisation strategies and distributed registration methodology are versatile and can be applied to a wide range of data that exhibit varying sensor qualities and properties. The development and application of these techniques does however lead to several further interesting opportunities and related remaining opens questions:

6.2.1 *Depth measurement resolution*

Firstly, our surface estimation model is currently built in a non-parametric fashion and we note that the expense of constructing such data-driven approaches typically grows with the magnitude of the available sample size. Point registration experiments in this thesis were carried out using relatively low resolution and low sample density point clouds compared to those that might be provided by *e.g.* professional high-end time-of-flight or triangulation based laser scanners capable of offering individual viewpoint measurements containing data on the order of millions of depth samples per view. Scaling our non-parametric models in a naive fashion is unlikely to prove a viable route to address this point. This would greatly increase the number of sample points contributing to a density estimate and, hence, likely lead to practical problems such as infeasible model construction and slow alignment quality evaluation during optimisation. In practice, early work attempting to reason about latent surface existence using extremely

high density point cloud evidence proved largely infeasible due to the computational demands of serially evaluating enormous data-driven density estimates.

In Chapter 3 we demonstrated one approach that could be employed to mitigate this problem; simple down-sampling of the available depth data. This strategy can potentially be applied at both the model building and alignment-evaluation (query time) stages. While simple down-sampling approaches were utilised in this work, further experimentation with more advanced sampling methods such as employing local spatial information (*e.g.* surface curvature) to inform sampling decisions offers an interesting and potentially straightforward extension for our non-parametric registration strategy. However while such down-sampling approaches work well in practice, they may ultimately result in an unpredictable loss of surface estimation quality typical of the reliability found under heuristic approximations.

A related conspicuous question that can be asked of our multi-view registration approach revolves around the proposed combination of solving large data set problem instances with kernel density estimation theory. While we have highlighted several of the benefits of this approach discovered and confirmed experimentally we concede that non-parametric methods are historically appropriate when sample sizes are small. When data sets become large, the central limit theorem states that sample means will follow a normal distribution, even if the respective variables are not normally distributed in the population. While they have been well-studied, non-parametric density estimation techniques in general tend to be expensive on massive datasets and it can be argued that parametric methods, which are typically more sensitive (*i.e.* have more statistical power), are in many cases the appropriate choice for large sample size problems.

In opposition to this point of view, increases in modern computational power motivate a growing trend to accept data-driven density estimation as essential statistical apparatus for large-scale data analysis, physical simulations and important tools for a broad variety of applications. Direct evidence in support of this trend is found in Chapter 4 where we demonstrate how to implement our density estimation ideas using the introduced distributed framework allowing for a practical solution that enables the exploitation of powerful non-parametric approaches for surface estimation in conjunction with large view set data. While we concede that the robustness of non-parametric methods come at the cost of requiring larger sample sizes to draw conclusions comparable to parametric approaches (*i.e.* with a matching degree of confidence) this cost allows

model anatomy to remain unspecified *a priori* and by instead determining structure from the data we provide an ability to remain highly flexible to arbitrary surface shape.

An alternative interesting and principled extension would be to introduce *approximate* density queries offering theoretical guarantees on the approximation-quality and time trade-off. Recent work (*e.g.* [286]) proposes an ability to return *approximated* density queries that would allow for an exploration of available trade-offs between speed and acceptable approximation error. Error could be quantitatively assessed by *e.g.* examining discrepancy between a full kernel density estimate and resulting approximations. However, it is not clear how these approximations should be chosen in order to meaningfully maintain a model's ability to *e.g.* represent high resolution surface detail (that in turn has an ability to aid registration) whilst introducing desirable properties such as frugal density query evaluation. Such exploration would enable further investigation of the desirable mutualism characteristics we find between viewpoint count and surface approximation reliability and quality.

6.2.2 *Global optimisation and objective function formulation*

While our models are able to handle the reconstruction of objects and environments exhibiting arbitrary geometrical complexity they currently contain no special handling of sharp features such as might be commonly found during the measurement of *e.g.* mechanical or machined parts. Related work addressing sharp features has been introduced by [114] and incorporating such considerations into our registration framework provide an additional avenue of future work.

The point set registration methods introduced in this thesis make use of local optimisation techniques, spatial transform parameters are optimised for each viewpoint independently (thus keeping the number of variables in each individual optimisation task low) yet the result of these procedures are iteratively used to update surface estimations, taking into consideration the individual movement (local optimisation) of each viewpoint. By iterating this process, the positions of all viewpoints are treated simultaneously and an ability to move views in the transform space at the same time is enabled. Employing this iterative, simultaneous, soft correspondence optimisation strategy with large view-set problem instances proves to be effective yet we concede that global maxima cannot be guaranteed, as is possible with true global optimisation

based approaches. In this sense, the strategies proposed in this work might be viewed as compromises; capable of producing solutions often in agreement with global optima in practice yet able to maintain run-time feasibility requirements on the demanding problems related to the registration of large view sets. Reasonable global optimisation algorithms may be applicable to the objective functions proposed in this work with relatively small additional overhead. The high dimensionality of global optimisations pertaining to a naive representation of large view-set problem instances would be an artefact of the problem representation yet interestingly an optimal registration might also be found in an intrinsically lower-dimensional global space. If, for example, many views are acquired from a sensor with a high temporal capture frame rate then the optimal alignment of views to a reference frame most likely occurs in a coordinated way. Exploring the exploitation of lower-dimensional global transform space manifolds for pose optimisation is an interesting direction for future work.

Additionally, it is conceded that the computational costs of registration-optimisation have not been formally treated here yet finding only locally optimal solutions is intuitively easier than searching for high dimensional global optima. We do not provide a rigorous mathematical definition or assurances about the basin of convergence of the proposed registration methods (many possible factors can influence convergence basin *shape* and *dimension*) however the registration experiments performed provide empirical evidence that the introduced approach is capable of handling moderate initial view misalignments *i.e.* starting conditions that are unfavourable, exhibiting greater misalignments than those we might normally classify as acceptable input to a global registration problem. The common mathematical framework of optimising a cost function is followed and while less attention has been focused on the optimisation method itself. There is a body of work in *e.g.* the 2D intensity image registration domain [136] that reports local optimisation methods are sometimes not sufficient to reliably find a global minima. Proposing a global optimisation method that is specifically tailored to the studied form of registration problem suggest an area for future work which would allow for progress with respect to both the question of *guaranteeing* global optima and that of exploring related lower-dimensional manifolds to search in.

6.2.3 *Real-time registration*

As highlighted in recent work [135, 29] the progression of commodity 3D range sensor capture rates has resulted in the ability to acquire partial 3D measurements of environments and objects becoming increasingly accessible and useful to a critical mass of practitioners. Contemporary, inexpensive end-user hardware capable of providing fast and abundant, yet typically noisy depth data, is now widely available. This explosion of data collection drives a need for large-scale *real-time* 3D point cloud processing and registration techniques. We claim that, with the development of such sensors, registering large numbers of range images and point clouds in real-time becomes of great interest and necessary for contemporary modelling pipelines. While seminal work such as KinectFusion [190] has made great inroads on this subtopic, many questions remain open such as how to handle objects that move (change their pose) or deform in real-time. While employing deformable registration methods (*e.g.* inspired by recent isometry-invariant correspondence [200] or 3D animation work) is an obvious starting point, making use of simultaneous registration to consider small *time windows of viewpoint information* rather than naive pairwise chain view alignment would prove an interesting avenue of exploration and might help to improve registration performance for time critical applications.

The introduced strategies provide the ability to *feasibly* perform effective *global optimisation* on huge viewpoint alignment problems and thus afford the advantages that effective global optimisation bring, however real-time applicability eludes the current methodology. This is due in part to both the strategy of iteratively improving surface estimates and our density estimation approach. As discussed real-time view-point registration is a highly attractive feature and progressing the introduced work in this direction would offer further valuable contributions. A practical, timely and attractive route to exploring real-time scenarios, able to combine with registration methodology developed in this work, would benefit from a successful application of GPU based methodology *e.g.* that found in [104, 190]. Such solutions are directly applicable to elements of our density estimation models *e.g.* parallelising registration components at finer granularities than the viewpoint level such as query-point density evaluation. Investigating algorithmic implementation on GPU architecture in practice provides an additional line of enquiry.

BIBLIOGRAPHY

[1] A.F. Abate, M. Nappi, D. Riccio, and G. Sabatino. 2D and 3D face recognition: A survey. *Pattern Recognition Letters*, 28(14):1885–1906, 2007.

[2] T. Abdelzaher, G. Thaker P., and Lardieri. A Feasible Region for Meeting Aperiodic End-to-End Deadlines in Resource Pipelines. In *Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS 2004)*, ICDCS 2004, pages 436–445, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 0-7695-2086-3. URL http://dl.acm.org/citation.cfm?id=977400.977975.

[3] I. S. Abramson. On Bandwidth Variation in Kernel Estimates-A Square Root Law. *The Annals of Statistics*, 10(4):1217–1223, 12 1982. doi: 10.1214/aos/1176345986. URL http://dx.doi.org/10.1214/aos/1176345986.

[4] V. Adve, R. Bagrodia, J. Browne, E. Deelman, A. Dubeb, E. Houstis, J. Rice, R. Sakellariou, D. Sundaram-Stukel, P. Teller, and M. Vernon. POEMS: End-to-end Performance Design of Large Parallel Adaptive Computational Systems. *Software Engineering*, 26(11):1027–1048, 2000.

[5] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building Rome in a Day. *Commun. ACM*, 54(10):105–112, October 2011. ISSN 0001-0782. doi: 10.1145/2001269.2001293. URL http://doi.acm.org/10.1145/2001269.2001293.

[6] B. Akinci, F. Boukamp, C. Gordon, D. Huber, C. Lyons, and K. Park. A formalism for utilization of sensor systems and integrated project models for active construction quality control. *Automation in Construction*, 15(2):124–138, 2006.

[7] D. Alexiadis, D. Zarpalas, and P. Daras. Real-time, full 3-D reconstruction of moving foreground objects from multiple consumer depth camera. *IEEE Transactions on Multimedia*, 15(2):339–358, February 2013.

[8] P. Alliez, L. Saboret, and G. Guennebaud. Surface reconstruction from point sets. Research manual – edition 3.5, Inria Sophia-Antipolis, 2009.

[9] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-D point sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 9 (5):698–700, 1987.

[10] ASUSTeK Computer Inc. ASUSTeK Computer Inc. http://www.asus.com/Multimedia/Xtion_PRO_LIVE/ Accessed February 2015.

[11] M. Audette, F. Ferrie, and T. Peters. An algorithmic overview of surface registration techniques for medical imaging. *Medical Image Analysis*, 3(4):201–217, 2000.

[12] M. Baker, B. Carpenter, and A. Shafi. MPJ express: Towards Thread Safe Java HPC, submitted to the. In *IEEE International Conference on Cluster Computing (Cluster 2006)*, pages 25–28, 2006.

[13] P. Bariya and K. Nishino. Scale-hierarchical 3D object recognition in cluttered scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1657–1664. IEEE, 2010.

[14] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *Computer Vision–ECCV 2006*, pages 404–417. Springer, 2006.

[15] R. Benjemaa and F. Schmitt. A Solution for the Registration of Multiple 3D Point Sets Using Unit Quaternions. In *Proceedings of the 5th European Conference on Computer Vision - Volume II*, ECCV '98, pages 34–50, London, UK, 1998. Springer-Verlag. ISBN 3-540-64613-2. URL http://dl.acm.org/citation.cfm?id=645312.648944.

[16] R. Benjemaa and F. Schmitt. Fast global registration of 3D sampled surfaces using a multi-Z-buffer technique. *Image Vision Comput.*, 17(2):113–123, 1999.

[17] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, pages 509–517, September . ISSN 0001-0782. doi: 10.1145/361002.361007.

[18] M. Berger, J. Levine, L. Nonato, G. Taubin, and C. Silva. An end-to-end framework for evaluating surface reconstruction. Technical report, Citeseer, 2011.

[19] R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau. Towards a general multiview registration technique. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(5):540–547, May 1996. ISSN 0162-8828. doi: 10.1109/34.494643. URL http://dx.doi.org/10.1109/34.494643.

[20] F. Bernardini and H. Rushmeier. The 3D model acquisition pipeline. In *Computer graphics forum*, volume 21:2, pages 149–172. Wiley Online Library, 2002.

[21] P. Besl and N. McKay. A method for registration of 3-D shapes. *IEEE. Trans. on Pattern Analysis and Machine Intelligence*, 14:239–256, 2 1992.

[22] P. Bhattacharyya and B. K. Chakrabarti. The mean distance to the n-th neighbour in a uniform distribution of random points: an application of probability theory. *European Journal of Physics*, 29(3):639–645, 2008. URL http://stacks.iop.org/0143-0807/29/i=3/a=023.

[23] L. Bin and E. R. Hancock. Structural graph matching using the EM algorithm and singular value decomposition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(10):1120–1136, 2001.

[24] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[25] G. Blais and M. D. Levine. Registering multiview range data to create 3D computer objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(8):820–824, Aug 1995. ISSN 0162-8828. doi: 10.1109/34.400574.

[26] F. Bonarrigo and A. Signoroni. An enhanced optimisation-on-a-manifold framework for global registration of 3D range data. *3D Image Modelling Proc. Vis. and Trans. 2011 (3DIMPVT).*, pages 350–357, 2011.

[27] F. Bonarrigo and A. Signoroni. Global registration of large collections of range images with an improved optimization-on-a-manifold approach. *Image and Vision Computing*, 32(6-7):437–451, 2014. ISSN 0262-8856. doi: http://dx.doi.org/10.1016/j.imavis.2014.02.012.

[28] O. Bonorden, B. Juurlink, I. Von Otte, and I. Rieping. The paderborn university BSP public library. *Parallel Computing*, 29(2):187–207, 2003.

[29] B. Boom, S. Orts-Escolano, X. X. Ning, S. McDonagh, P. Sandilands, and R. B. Fisher. Point light source estimation based on scenes recorded by a RGB-D camera. In *British Machine Vision Conference*, Sept 2013.

[30] D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, and J. Hertzberg. Globally Consistent 3D Mapping with Scan Matching. *Robot. Auton. Syst.*, 56(2):130–142, feb 2008. ISSN 0921-8890. doi: 10.1016/j.robot.2007.07.002. URL http://dx.doi.org/10.1016/j.robot.2007.07.002.

[31] K. Bowyer, K. Chang, and P. Flynn. A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition. *Computer Vision and Image Understanding, CVIU*, 101(1):1–15, 2006.

[32] L. Breiman, W. Meisel, and E. Purcell. Variable Kernel Estimates of Multivariate Densities. *Technometrics*, 19(2):135–144, 1977.

[33] D. Breitenreicher and C. Schnörr. Intrinsic Second-Order Geometric Optimization for Robust Point Set Registration Without Correspondence. In *Proceedings of the 7th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*, EMMCVPR '09, pages 274–287, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-03640-8. doi: 10.1007/978-3-642-03641-5_21. URL http://dx.doi.org/10.1007/978-3-642-03641-5_21.

[34] D. Breitenreicherand and C. Schnörr. Model-Based Multiple Rigid Object Detection and Registration in Unstructured Range Data. *Int. J. Comput. Vision*, 92(1):32–52, mar 2011. ISSN 0920-5691. doi: 10.1007/s11263-010-0401-3. URL http://dx.doi.org/10.1007/s11263-010-0401-3.

[35] Brekel Kinect scanner. Brekel Kinect scanner. http://brekel.com/kinect-3d-scanner/ Accessed February 2015.

[36] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Three-dimensional face recognition. *International Journal of Computer Vision, IJCV.*, 64(1):5–30, 2005.

[37] L. G. Brown. A Survey of Image Registration Techniques. *ACM Comput. Surv.*, 24(4):325–376, Dec 1992. ISSN 0360-0300. doi: 10.1145/146370.146374. URL http://doi.acm.org/10.1145/146370.146374.

[38] M. Buehler, K. Iagnemma, and S. Singh. The 2005 DARPA grand challenge. *Springer Tracts in Advanced Robotics*, 36(5):1–43, 2007.

[39] A. Bugeau and P. Perez. Bandwidth selection for kernel estimation in mixed multi-dimensional spaces. Research Report INRIA. no. 6286, INRIA, 2007.

[40] R. Buyya, D. Abramson, and J. Giddy. Nimrod/G: An architecture for a resource management and scheduling system in a global computational grid. In *High Performance Computing in the Asia-Pacific Region. Proceedings. The Fourth International Conference on*, volume 1, pages 283–289, May 2000.

[41] R. Buyya, M. Murshed, and D. Abramson. A Deadline and Budget Constrained Cost-Time Optimisation Algorithm for Scheduling Task Farming Applications on Global Grids. Technical report, Monash University, March 2002.

[42] N. D. F. Campbell, G. Vogiatzis, C. Hernandez, and R. Cipolla. Using Multiple hypotheses to Improve Depth-Maps for Multi-View Stereo. In David Forsyth, Philip Torr, and Andrew Zisserman, editors, *Cmputer Vision - ECCV 2008*, volume 5302 of *Lecture Notes in Computer Science*, pages 766–779. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-88681-5. doi: 10.1007/978-3-540-88682-2_58. URL http://dx.doi.org/10.1007/978-3-540-88682-2_58.

[43] R. J. Campbell and P. J. Flynn. A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding, CVIU.*, 81 (2):166–210, 2001.

[44] H. Casanova, M. Kim, J. S. Plank, and J. Dongarra. Adaptive Scheduling for Task Farming with Grid Middleware. *International Journal of High Performance Computing*, 13(3):231–240, August 1999.

[45] H. Casanova, G. Obertelli, F. Berman, and R. Wolski. The AppLeS Parameter Sweep Template: User-level Middleware for the Grid. In *Proceedings of the 2000 ACM/IEEE Conference on Supercomputing*, Supercomputing '00, Washington,

DC, USA, 2000. IEEE Computer Society. ISBN 0-7803-9802-5. URL http://dl.acm.org/citation.cfm?id=370049.370499.

[46] U. Castellani, A. Fusiello, and V. Murino. Registration of Multiple Acoustic Range Views for Underwater Scene Reconstruction. *Computer Vision and Image Understanding*, 87(1-3):78–89, 2002. ISSN 1077-3142. doi: http://dx.doi.org/10.1006/cviu.2002.0984. URL http://www.sciencedirect.com/science/article/pii/S1077314202909847.

[47] U. Castellani, A. Fusiello, V. Murino, L. Papaleo, E. Puppo, and M. Pittore. A complete system for on-line 3D modelling from acoustic images. *Signal Processing: Image Communication*, 20(9):832–852, 2005.

[48] U. Castellani, V. Gay-Bellile, and A. Bartoli. Robust deformation capture from temporal range data for surface rendering. *Computer Animation and Virtual Worlds*, 19(5):591–603, 2008.

[49] J. E. Chacon and T. Duong. Multivariate plug-in bandwidth selection with unconstrained pilot bandwidth matrices. *TEST*, 19(2):375–398, 2010.

[50] C. Chen and I. Stamos. Semi-automatic range to range registration: a feature-based method. In *3D Digital Imaging and Modeling, 2005. 3DIM 2005. Fifth International Conference on*, pages 254–261, June 2005. doi: 10.1109/3DIM.2005.72.

[51] C. Chen, Y. Hung, and J. Cheng. A fast automatic method for registration of partially-overlapping range images. In *Computer Vision, 1998. Sixth International Conference on*, pages 242–248. IEEE, 1998.

[52] C. Chen, Y. Hung, and J. Cheng. RANSAC-based DARCES: A new approach to fast automatic registration of partially overlapping range images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(11):1229–1234, 1999.

[53] T. Chen, B. C. Vemuri, A. Rangarajan, and S. J. Eisenschenk. Group-wise Point-set registration using a novel CDF-based Havrda-Charvat Divergence. *International Journal of Computer Vision*, 86(1):111–124, Jan 2010.

[54] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Int. Journal Computer Vision and Image Understanding (IJCVU).*, 3 (10):145–155, 1992.

[55] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek. The trimmed iterative closest point algorithm. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 3, pages 545–548 vol.3, 2002.

[56] H. Chui and A. Rangarajan. A feature registration framework using mixture models. In *Mathematical Methods in Biomedical Image Analysis, 2000. Proceedings. IEEE Wrkshop on*, pages 190–197. IEEE, 2000.

[57] H. Chui and A. Rangarajan. A new algorithm for non-rigid point matching. In *Computer Vision and Pattern Recognition, 2000. CVPR 2000. Proceedings of the 2000 IEEE Computer Society Conference on*, volume 2, pages 44–51, 2000.

[58] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89:114–141, 2003.

[59] H. Chui, L. Win, R. Schultz, J. S. Duncan, and A. Rangarajan. A unified non-rigid feature registration method for brain mapping. *Medical Image Analysis*, 7(2):113–130, 2003. ISSN 1361–8415. doi: http://dx.doi.org/10.1016/S1361-8415(02)00102-0. URL http://www.sciencedirect.com/science/article/pii/S1361841502001020. Mathematical Methods in Biomedical Image Analysis – MMBIA 2001.

[60] L. Clements, W. C. Chapman, B. M. Dawant, R. L. Galloway, and M. I. Miga. Robust surface registration using salient anatomical features for image-guided liver surgery: algorithm and validation. *Medical Physics*, 35(6):2528–2540, 2008.

[61] M. Cole. *Algorithmic skeletons: structured management of parallel computation*. MIT Press, Cambridge, MA, USA, 1991. ISBN 0-262-53086-4.

[62] D. Comaniciu, V. Ramesh, and P. Meer. The variable bandwidth mean shift and data-driven scale selection. In *Computer Vision, ICCV. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 438–445, 2001.

[63] G. M. Cortelazzo, G. Doretto, and L. Lucchese. Free-form textured surfaces registration by a frequency domain technique. In *Image Processing, 1998. ICIP*

98. Proceedings. 1998 International Conference on, volume 1, pages 813–817 vol.1, Oct 1998. doi: 10.1109/ICIP.1998.723634.

[64] A. D. J. Cross and E. R Hancock. Graph matching with a dual-step EM algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(11):1236–1253, 1998.

[65] S. Cunnington and A. J. Stoddart. N-View point set registration: A comparison. *British Machine Vision Conference*, pages 234–244, 1999.

[66] R. H. Davies, C. J. Twining, T. F. Cootes, J. C. Waterton, and C. J. Taylor. A minimum description length approach to statistical shape modeling. *IEEE Trans Med Imaging*, 21(5):525–537, May 2002.

[67] H. de Ruiter and B. Benhabib. On-line modeling for real-time, model-based, 3D pose tracking. In *Advances and Innovations in Systems, Computing Sciences and Software Engineering*, pages 555–560. Springer, 2007.

[68] J. Dean and S. Ghemawat. MapReduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, Jan 2008. ISSN 0001-0782. doi: 10.1145/1327452.1327492. URL http://doi.acm.org/10.1145/1327452.1327492.

[69] P. Deheuvels. Estimation non parametrique de la densite par histogrammes generalises. *Revue de Statistique Appliquee*, 25(3):5–42, 1977. URL http://eudml.org/doc/106046.

[70] D. Devroye, J. Beirlant, R. Cao, R. Fraiman, P. Hall, M. C. Jones, G. Lugosi, E. Mammen, J. S. Marron, C. Sanchez-Sellero, J. Una, F. Udina, and L. Devroye. Universal smoothing factor selection in density estimation: theory and practice. *TEST*, 6(2):223–320, 1997.

[71] L. Devroye and G. Lugosi. Variable kernel estimates: on the impossibility of tuning the parameters. In *High-Dimensional Probability II*, pages 405–424, New York, 2000. Springer-Verlag.

[72] G. Dewaele, F. Devernay, and R. Horaud. Hand motion from 3D point trajectories and a smooth surface model. In Pajdla T. and Matas J., editors, *The 8th European Conference on Computer Vision (ECCV 2004)*, volume 3021 of *Lecture Notes in*

Computer Science, pages 495–507. Springer Berlin Heidelberg, 2004. ISBN 978-3-540-21984-2. doi: 10.1007/978-3-540-24670-1_38. URL http://dx.doi.org/10.1007/978-3-540-24670-1_38.

[73] Dimensional Imaging. Dimensional Imaging. http://www.di3d.com/ Accessed February 2015.

[74] L. Ding, A. Goshtasby, and M. Satter. Volume image registration by template matching. *Image and Vision Computing*, 19(12):821–832, 2001. ISSN 0262-8856. doi: http://dx.doi.org/10.1016/S0262-8856(00)00101-3. URL http://www.sciencedirect.com/science/article/pii/S0262885600001013.

[75] C. Dorai, J. Weng, and A. K. Jain. Optimal registration of object views using range data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(10):1131–1138, 1997.

[76] H. Du, P. Henry, X. Ren, M. Cheng, D. B. Goldman, S. M. Seitz, and D. Fox. Interactive 3D modeling of indoor environments with a consumer depth camera. In *Proceedings of the 13th International Conference on Ubiquitous computing*, pages 75–84. ACM, 2011.

[77] T. Duong. KS: Kernel Density Estimation and Kernel Discriminant Analysis for Multivariate Data in R. *Journal of statistical software*, 21(7):1–16, 01 2007. http://www.jstatsoft.org/v21/i07/bibtex.

[78] T. Duong and M. L. Hazelton. Cross-validation Bandwidth Matrices for Multivariate Kernel Density Estimation. *Scandinavian Journal of Statistics*, 32(3):485–506, 2005. ISSN 1467-9469. doi: 10.1111/j.1467-9469.2005.00445.x. URL http://dx.doi.org/10.1111/j.1467-9469.2005.00445.x.

[79] T. Duong and M. L. Hazelton. Convergence rates for unconstrained bandwidth matrix selectors in multivariate kernel density estimation. *Journal of Multivariate Analysis*, 93(2):417–433, 2005.

[80] S. Durrleman, X. Pennec, A. Trouve, and N. Ayache. Statistical models of sets of curves and surfaces based on currents. *Medical Image Analysis*, 13(5):793–808, Oct 2009.

[81] D. L. Eager, J. Zahorjan, and E. D. Lazowska. Speedup versus efficiency in parallel systems. *IEEE Transactions on Computers*, 38(3):408–423, 1989. ISSN 0018-9340. doi: 10.1109/12.21127.

[82] ECDF. The Edinburgh Compute and Data Facility. http://www.wiki.ed.ac.uk/display/ecdfwiki/ Accessed February 2015.

[83] D. W. Eggert, A. W. Fitzgibbon, and R. B. Fisher. Simultaneous registration of multiple range views for use in reverse engineering. *Proc. of the 13th Int. Conference on Pattern Recognition.*, pages 243–247, 1996.

[84] D. W. Eggert, A. W. Fitzgibbon, and R. B. Fisher. Simultaneous registration of multiple range views for use in reverse engineering of CAD models. *Computer Vision and Image Understanding*, 69(3):253–272, 1998.

[85] W. R. Elwasif, J. S. Plank, and R. Wolski. Data Staging Effects in Wide Area Task Farming Applications. In *Proceedings of IEEE International Symposium on Cluster Computing and the Grid*, Brisbane, Australia, May 2001.

[86] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard. An evaluation of the RGB-D SLAM system. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1691–1696. IEEE, 2012.

[87] G. D. Evangelidis, D. Kounades-Bastian, R. Horaud, and E. Z. Psarakis. A Generative Model for the Joint Registration of Multiple Point Sets. In *Computer Vision – ECCV 2014*, pages 109–122. Springer, 2014.

[88] S. Fantoni and U. Castellani. Accurate and automatic alignment of range surfaces. *3D Image Modelling Proc. Vis. and Trans 2012 (3DIMPVT).*, pages 73–80, 2012.

[89] M. Faugeras and M. Herbert. Representation, recognition and locating of 3D objects. *International Conference on Pattern Recognition*, 1986.

[90] J. Feldmar and N. Ayache. Rigid, affine and locally affine registration of free-form surfaces. *International Journal of Computer Vision*, 18(2):99–119, 1996.

[91] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography.

Commun. ACM, 24(6):381–395, June 1981. ISSN 0001-0782. doi: 10.1145/358669.358692. URL http://doi.acm.org/10.1145/358669.358692.

[92] R. B. Fisher. Applying knowledge to reverse engineering problems. *Computer-Aided Design*, 36(6):501–510, 2004.

[93] R. B. Fisher, T. P. Breckon, K. Dawson-Howe, A. Fitzgibbon, C. Robertson, E. Trucco, and C. K. I. Williams. *Dictionary of computer vision and image processing*. John Wiley & Sons, 2013.

[94] A. W. Fitzgibbon. Robust registration of 2D and 3D point sets. *Image and Vision Computing*, 21(13):1145–1153, 2003.

[95] L. M. G. Fonseca and B. S. Manjunath. Registration Techniques for Multisensor Remotely Sensed Imagery. *Journal of Photogrammetry Engineering and Remote Sensing*, 62(9):1049–1056, Sep 1996.

[96] D. A. Forsyth and J. Ponce. *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference, 2002.

[97] I. Foster. Task Parallelism and High-Performance Languages. *IEEE Parallel Distrib. Technol.*, 2(3):27–36, Sep 1994. ISSN 1063-6552. doi: 10.1109/M-PDT.1994.329794. URL http://dx.doi.org/10.1109/M-PDT.1994.329794.

[98] J. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys. Building rome on a cloudless day. In Daniilidis K., Maragos P., and Paragios N., editors, *Computer Vision - ECCV 2010*, volume 6314 of *Lecture Notes in Computer Science*, pages 368–381. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-15560-4. doi: 10.1007/978-3-642-15561-1_27. URL http://dx.doi.org/10.1007/978-3-642-15561-1_27.

[99] M. Franaszek, G. S. Cheok, K. S. Saidi, and C. Witzgall. Fitting spheres to range data from 3-D imaging systems. *Instrumentation and Measurement, IEEE Transactions on*, 58(10):3544–3553, 2009.

[100] M. I. Frank, A. Agarwal, and M. K. Vernon. LoPC: modeling contention in parallel algorithms. In *Proceedings of the sixth ACM SIGPLAN symposium on*

Principles and practice of parallel programming, PPOPP 1997, pages 276–287, New York, NY, USA, 1997. ACM. ISBN 0-89791-906-8.

[101] T. Funkhouser and M. Kazhdan. Shape-based retrieval and analysis of 3D models. In *ACM SIGGRAPH 2004 Course Notes*, page 16. ACM, 2004.

[102] A. Fusiello. Visione computazionale. lecture notes. *Appunti delle lezioni. Pubblicato a cura dell'autore*, 2008.

[103] H. Gagnon, M. Soucy, R. Bergevin, and D. Laurendeau. Registration of multiple range views for automatic 3D model building. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR 1994. IEEE Computer Society Conference on*, pages 581–586. IEEE, 1994.

[104] V. Garcia, E. Debreuve, and M. Barlaud. Fast K-nearest neighbor search using GPU. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–6. IEEE, 2008.

[105] W. Gentzsch. Sun Grid Engine: Towards Creating a Compute Power Grid. In *Proceedings of the 1st International Symposium on Cluster Computing and the Grid*, CCGRID 2001, pages 35–43, Washington, DC, USA, 2001. IEEE Computer Society. ISBN 0-7695-1010-8. URL http://dl.acm.org/citation.cfm?id=560889.792378.

[106] S. Gold, A. Rangarajan, C. Lu, and E. Mjolsness. New Algorithms for 2D and 3D Point Matching: Pose Estimation and Correspondence. *Pattern Recognition*, 31:957–964, 1997.

[107] C. Goldfeder, M. Ciocarlie, J. Peretzman, H. Dang, and P. K. Allen. Data-driven grasping with partial sensor data. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 1278–1283. IEEE, 2009.

[108] M. W. Goudreau, K. Lang, S. B. Rao, T. Suel, and T. Tsantilas. Portable and efficient parallel computing using the bsp model. *Computers, IEEE Transactions on*, 48(7):670–689, 1999.

[109] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 1994. ISBN 0201558025.

[110] S. Granger and X. Pennec. Multi-scale EM-ICP: A Fast and Robust Approach for Surface Registration. In *European Conference on Computer Vision (ECCV 2002), volume 2353 of LNCS*, pages 418–432. Springer, 2002.

[111] S. Granger, X. Pennec, and A. Roche. Rigid point-surface registration using an EM variant of ICP for computer guided oral implantology. *Int. Conference on Medical Image Comp. and Comp. Assisted Intervention*, 4:752–761, 2001.

[112] A. Gray and A. Moore. Rapid Evaluation of Multiple Density Models. In *Artificial Intelligence and Statistics*, 2003.

[113] M. Greenspan and P. Boulanger. Efficient and reliable template set matching for 3d object recognition. In *3-D Digital Imaging and Modeling, 1999. Proceedings. Second International Conference on*, pages 230–239, 1999. doi: 10.1109/IM.1999.805353.

[114] A. Haim, A. Sharf, C. Greif, and D. Cohen-Or. L1-Sparse reconstruction of sharp point set surfaces. *ACM Trans. Graphics*, 29(5):135–147, 2010.

[115] S. D. Hammond, J. A. Smith, G. R. Mudalige, and S. A. Jarvis. Predictive Simulation of HPC Applications. In *The IEEE 23rd International Conference on Advanced Information Networking and Applications (AINA-09)*, 2009.

[116] S. D. Hammond, G. R. Mudalige, J. A. Smith, S. A. Jarvis, J. A. Herdman, and A. Vadgama. WARPP: A toolkit for simulating high-performance parallel scientific codes. In *2nd International ICST Conference on Simulation Tools and Techniques*. ACM, 5 2010. doi: 10.4108/ICST.SIMUTOOLS2009.5753.

[117] J. Han, L. Shao, D. Xu, and J. Shotton. Enhanced Computer Vision with Microsoft Kinect Sensor: A Review. *IEEE Trans. Cybernetics*, 43(5), October 2013. URL http://research.microsoft.com/apps/pubs/default.aspx?id=194894.

[118] W. Hardle. *Smoothing Techniques with Implementations in S*. Springer-Verlag. Springer Series in Statistics, New-York, 1991.

[119] W. Hardle, M. Muller, S. Sperlich, and A. Werwatz. *Nonparametric and Semi-parametric Models*. Springer-Verlag. Springer Series in Statistics, 2004.

[120] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[121] Y. He and Y. Mei. An efficient registration algorithm based on spin image for lidar 3D point cloud models. *Neurocomputing*, pages 1–10, 2014. ISSN 0925-2312. doi: http://dx.doi.org/10.1016/j.neucom.2014.09.029. URL http://www.sciencedirect.com/science/article/pii/S0925231214012120.

[122] N. B. Heidenreich, A. Schindler, and S. Sperlich. Bandwidth selection for kernel density estimation: a review of fully automatic selectors. *Advances in Statistical Analysis*, 97(4):403–433, 2013.

[123] G. Hetzel, B. Leibe, P. Levi, and B. Schiele. 3D Object Recognition from Range Images using Local Feature Histograms. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, December 2001.

[124] D. L. G. Hill, P. G. Batchelor, M. Holden, and D. J. Hawkes. Medical image registration. *Physics in Medicine and Biology*, 46(3):R1, 2001. URL http://stacks.iop.org/0031-9155/46/i=3/a=201.

[125] J. Hill, B. McColl, D. C. Stefanescu, M. W. Goudreau, K. Lang, S. B. Rao, T. Suel, T. Tsantilas, and R. H. Bisseling. BSPlib: The BSP programming library. *Parallel Computing*, 24(14):1947–1980, 1998.

[126] J. M. D. Hill, S. Donaldson, and A. McEwan. Installation and User Guide for the Oxford BSP toolset implementation of bsplib. 1997.

[127] W. D. Hillis, J. Steele, and L. Guy. Data Parallel Algorithms. *Commun. ACM*, 29(12):1170–1183, Dec 1986. ISSN 0001-0782. doi: 10.1145/7902.7903. URL http://doi.acm.org/10.1145/7902.7903.

[128] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4(4):629–642, 1987.

[129] Q. Huang, B. Adams, and M. Wand. Bayesian surface reconstruction via iterative scan alignment to an optimized prototype. In *Symposium on Geometry Processing*, pages 213–223, 2007.

[130] D. F. Huber. *Automatic Three-dimensional Modeling from Reality*. PhD thesis, The Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, July 2002.

[131] D. F. Huber and M. Hebert. Fully automatic registration of multiple 3D data sets. *Image and Vision Computing*, 21:637–650, 2001.

[132] K. Ikeuchi, A. Nakazawa, K. Hasegawa, and T. Ohishi. The Great Buddha Project: Modeling Cultural Heritage for VR Systems Through Observation. In *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '03, pages 7–, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-2006-5. URL http://dl.acm.org/citation.cfm?id=946248.946860.

[133] K. Ikeuchi, T. Oishi, J. Takamatsu, R. Sagawa, A. Nakazawa, R. Kurazume, K. Nishino, M. Kamakura, and Y. Okamoto. The Great Buddha Project: Digitally archiving, restoring, and analyzing cultural heritage objects. *Int. Journal of Computer Vision*, 75(1):189–208, 2007.

[134] M. Isard, M. Budiuand, Y. Yu, A. Birrell, and D. Fetterly. Dryad: distributed data-parallel programs from sequential building blocks. In *Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007*, EuroSys '07, pages 59–72, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-636-3. doi: 10.1145/1272996.1273005. URL http://doi.acm.org/10.1145/1272996.1273005.

[135] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, et al. KinectFusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011.

[136] M. Jenkinson and S. Smith. A global optimisation method for robust affine registration of brain images. *Medical image analysis*, 5(2):143–156, 2001.

[137] B. Jian and B. C. Vemuri. Robust Point Set Registration Using Gaussian Mixture Models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33 (8):1633–1645, Aug 2011. ISSN 0162-8828. doi: 10.1109/TPAMI.2010.223.

[138] H. Jin, S. Soatto, and A. J. Yezzi. Multi-view stereo reconstruction of dense shape and complex appearance. *International Journal of Computer Vision*, 63 (3):175–189, 2005.

[139] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):433–449, 1999.

[140] M. C. Jones, J. S. Marron, and S. J. Sheather. A Brief Survey of Bandwidth Selection for Density Estimation. *Journal of the American Statistical Association*, 91(433):401–407, 1996.

[141] M. C. Jones, J. S. Marron, and S. J. Sheather. Progress in data-based bandwidth selection for kernel density estimation. *Computational Statistics*, pages 337–381, 1996.

[142] A. Joshi and C. Lee. On the problem of correspondence in range data and some inelastic uses for elastic nets. *Neural Networks, IEEE Transactions on*, 6(3): 716–723, 1995.

[143] G. Kamberova and R. Bajcsy. Sensor errors and the uncertainties in stereo reconstruction. In *Workshop on Empirical Evaluation Methods in Computer Vision, Santa Barbara, California*. Citeseer, 1998.

[144] B. G. Kashef and A. A. Sawetauk. A survey of new techniques for image registration and mapping. In *Proc. SPIE*, volume 0432, pages 222–239, 1984. doi: 10.1117/12.936668. URL http://dx.doi.org/10.1117/12.936668.

[145] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. *SGP Proc. of the fourth Eurographics symposium on Geometry processing*, pages 61–70, 2006.

[146] D. J. Kerbyson, A. Hoisie, and H. J. Wasserman. Use of predictive performance modeling during large-scale system installation. *Parallel Processing Letters*, 15 (04):387–395, 2005. doi: 10.1142/S0129626405002301.

[147] S. Khoualed, U. Castellani, and A. Bartoli. Semantic shape context for the registration of multiple partial 3D views. *IEEE Transactions on pattern analysis and machine intelligence*, 14:239–256, 2009.

[148] S. Kim, C. Jho, and H. Hong. Automatic registration of 3D data sets from unknown viewpoints. In *Workshop on Frontiers of Computer Vision. FCV.*, pages 155–159, 2003.

[149] Kinect SDK. Kinect SDK. http://www.microsoft.com/en-us/kinectforwindows/ Accessed February 2015.

[150] K. Kolev, M. Klodt, T. Brox, S. Esedoglu, and D. Cremers. Continuous global optimization in multiview 3D reconstruction. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 441–452. Springer, 2007.

[151] K. Kolev, M. Klodt, T. Brox, and D. Cremers. Continuous global optimization in multiview 3d reconstruction. *International Journal of Computer Vision*, 84(1): 80–96, 2009.

[152] I. Kostrikov, E. Horbert, and B. Leibe. Probabilistic labeling cost for high-accuracy multi-view reconstruction. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1534–1541. IEEE, 2014.

[153] S. Krishnan, P. Y. Lee, J. B. Moore, and S. Venkatasubramanian. Global registration of multiple 3D point sets via optimisation-on-a-manifold. In *Symposium on Geometry Processing*, pages 187–196, 2005.

[154] S. Krishnan, P. Y. Lee, J. B. Moore, and S. Venkatasubramanian. Optimisation-on-a-manifold for global registration of multiple 3D point sets. *Int. J. Intell. Syst. Technol. Appl.*, 3(4):319–340, 2007.

[155] J. Labarta, S. Girona, and T. Cortes. Analysing Scheduling Policies using DIMEMAS. In *Parallel Computing*, number 1 in 23, April 1997.

[156] H. Lester and S. R. Arridge. A survey of hierarchical non-linear medical image registration. *Pattern Recognition*, 32(1):129–149, 1999. ISSN 0031-3203. doi: http://dx.doi.org/10.1016/S0031-3203(98)00095-8. URL http://www.sciencedirect.com/science/article/pii/S0031320398000958.

[157] H. Li, R. W. Sumner, and M. Pauly. Global Correspondence Optimization for Non-Rigid Registration of Depth Scans. *Computer Graphics Forum*, 27(5):1421–1430, 2008.

[158] Y. Liu. Automatic registration of overlapping 3D point clouds using closest points. *Image and Vision Computing*, 24(7):762–781, 2006.

[159] Y. Liu. Automatic range image registration in the markov chain. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(1):12–29, 2010.

[160] C. Loader. Bandwidth selection: Classical or plug-in? *The Annals of Statistics*, 27(2):415–781, 1999.

[161] D. O. Loftsgaardenand and C. P. Quesenberry. A Nonparametric Estimate of a Multivariate Density Function. *The Annals of Mathematical Statistics*, 36(3): 1049–1051, 6 1965.

[162] A. Lorusso, D. W. Eggert, and R. B. Fisher. A comparison of four algorithms for estimating 3-D rigid transformations. In *British Machine Vision Conference*. BMVC, 1995.

[163] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. IEEE, 1999.

[164] B. Luo and E. R. Hancock. A unified framework for alignment and correspondence. *Computer Vision and Image Understanding*, 92(1):26–55, 2003.

[165] Y. Ma. *An invitation to 3D vision: from images to geometric models*, volume 26. Springer Science & Business Media, 2004.

[166] Y. P. Mack and M. Rosenblatt. Multivariate k-nearest neighbor density estimates. *Journal of Multivariate Analysis*, 9(1):1–15, 1979. ISSN 0047-259X. doi: http://dx.doi.org/10.1016/0047-259X(79)90065-4.

[167] J. B. A. Maintz and M. A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, 1998. ISSN 1361-8415. doi: http://dx.doi.org/10.1016/S1361-8415(01)80026-8. URL http://www.sciencedirect.com/science/article/pii/S1361841501800268.

[168] A. Makadia, A. Patterson, and K. Daniilidis. Fully automatic registration of 3D point clouds. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 1297–1304. IEEE, 2006.

[169] J. S. Marron and D. Nolan. Canonical kernels for density estimation. *Statistics & Probability Letters*, 7(3):195–199, 1988.

[170] J. L. Martinez, A. J. Reina, J. Morales, A. Mandow, and A. J. Garcia-Cerezo. Using multicore processors to parallelize 3D point cloud registration with the coarse binary cubes method. In *Mechatronics (ICM), 2013 IEEE International Conference on*, pages 335–340. IEEE, 2013.

[171] T. Masuda. Generation of geometric model by registration and integration of multiple range images. In *3D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 254–261. IEEE, 2001.

[172] T. Masuda. Object shape modelling from multiple range images by matching signed distance fields. In *3D Data Processing Visualization and Transmission, 2002. Proceedings. First International Symposium on*, pages 439–448. IEEE, 2002.

[173] T. Masuda and N. Yokoya. A Robust Method for Registration and Segmentation of Multiple Range Images. *Computer Vision and Image Understanding*, 61(3):295–307, 1995. ISSN 1077-3142. doi: http://dx.doi.org/10.1006/cviu.1995.1024. URL http://www.sciencedirect.com/science/article/pii/S1077314285710247.

[174] Mathworks Inc. Mathworks Inc. http://www.mathworks.co.uk/ Accessed February 2015.

[175] W. F. McColl. General purpose parallel computing. *Lectures on Parallel Computation. Cambridge International Series on Parallel Computation*, pages 337–391, 1993.

[176] S. McDonagh and R. B. Fisher. Simultaneous registration of multi-view range images with adaptive kernel density estimation. In *14th IMA Conference on Mathematics of Surfaces*, volume 14, pages 31–62, Birmingham, September 2013.

[177] S. McDonagh, R. B. Fisher, and J. Rees. Using 3D information for classification of non-melanoma skin lesions. In *Proc. Medical Image Understanding and Analysis*, pages 164–168, Dundee, 2008.

[178] S. McDonagh, C. Beyan, P. X. Huang, and R. B. Fisher. Applying semi-synchronised task farming to large-scale computer vision problems. *International*

Journal of High Performance Computing Applications, pages 1–24, 2014. doi: 10.1177/1094342014532965. URL http://hpc.sagepub.com/content/early/2014/05/13/1094342014532965.abstract.

[179] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: a survey. *Medical image analysis*, 1(2):91–108, 1996.

[180] G. McNeill and S. Vijayakumar. A probabilistic approach to robust shape matching. In *Image Processing, 2006 IEEE International Conference on*, pages 937–940. IEEE, 2006.

[181] A. S. Mian, M. Bennamoun, and R. A. Owens. Automatic correspondence for 3D modeling: an extensive review. *International Journal of Shape Modeling*, 11(2): 253–291, 2005.

[182] A. S. Mian, M. Bennamoun, and R. Owens. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(10):1584–1601, 2006.

[183] Microsoft Kinect. Microsoft Corp. http://research.microsoft.com/en-us/projects/kinectforwindows/ Accessed February 2015.

[184] N. J. Mitra, S. Flöry, M. Ovsjanikov, N. Gelfand, L. J. Guibas, and H. Pottmann. Dynamic geometry registration. In *Symposium on Geometry Processing*, pages 173–182, 2007.

[185] A. Mittal and N. Paragios. Motion-based background subtraction using adaptive kernel density estimation. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–302–II–309 Vol.2, June 2004.

[186] J. Le Moigne, X. Wei, P. Chalermwat, T. El-Ghazawi, M. Mareboyana, N. Netanyahu, J. C. Tilton, W. J. Campbell, and R. P. Cromp. First evaluation of automatic image registration methods. In *Geoscience and Remote Sensing Symposium Proceedings, IGARSS 98 IEEE*, volume 1, pages 315–317, Jul 1998. doi: 10.1109/IGARSS.1998.702890.

[187] G. R. Mudalige, M. K. Vernon, and S. A. Jarvis. A plug-and-play model for evaluating wavefront computations on parallel architectures. In *Parallel and Dis-*

tributed Processing, 2008. IPDPS 2008. IEEE International Symposium on, pages 1–14, 2008. doi: 10.1109/IPDPS.2008.4536243.

[188] A. Myronenko and X. Song. Point set registration: Coherent Point Drift. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(12):2262–2275, 2010.

[189] P. J. Neugebauer. Reconstruction of Realworld Objets via Simultaneous Registration and Robust Combination of Multiple Range Images. In *International Journal of Shape Modeling*, number 1 in 3, pages 71–90. World Scientific Publishing Company, 1997.

[190] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. W. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE International symposium on*, pages 127–136. IEEE, 2011.

[191] G. R. Nudd, D. Kerbyson, E. Papaefstathiou, S. Perry, J. Harper, and D. Wilcox. PACE: A toolset for the performance prediction of parallel and distributed systems. *International Journal High Performance Computing Applications*, 14(03): 228–251, 2000.

[192] OSU database. Ohio State University. Range Image Collection. http://sampl.ece.ohio-state.edu/data/ Accessed April 2013.

[193] J. Orava. K-nearest neighbour kernel density estimation the choice of optimal k. *Tatra Mountains Mathematical Publications*, 50(1):39–50, Nov 2012. doi: 10.2478/v10127-011-0035-z.

[194] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Transactions on Graphics (TOG)*, 21(4):807–832, 2002.

[195] B. Park and B. Turlach. Practical performance of several data driven bandwidth selectors. CORE discussion papers, Universite catholique de Louvain, Center for Operations Research and Econometrics (CORE), 1992.

[196] I. K. Park, M. Germann, M. Breitenstein, and H. Pfister. Fast and automatic object pose estimation for range images on the GPU. *Machine Vision and Applications*, 21(5):749–766, 2010.

[197] E. Parzen. On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 09 1962. doi: 10.1214/aoms/1177704472. URL http://dx.doi.org/10.1214/aoms/1177704472.

[198] K. Pathak, A. Birk, N. Vaskevicius, M. Pfingsthorn, S. Schwertfeger, and J. Poppinga. Online three-dimensional slam by registration of large planar surface segments and closed-form pose-graph relaxation. *Journal of Field Robotics*, 27(1): 52–84, 2010.

[199] M. Pauly, M. Gross, and L. P. Kobbelt. Efficient Simplification of Point-sampled Surfaces. In *Proceedings of the Conference on Visualization '02*, VIS '02, pages 163–170, Washington, DC, USA, 2002. IEEE Computer Society. ISBN 0-7803-7498-3. URL http://dl.acm.org/citation.cfm?id=602099.602123.

[200] N. Pears, Y. Liu, and P. Bunting. *3D Imaging, Analysis and Applications*. Springer, 2012.

[201] X. Pennec. Multiple Registration and Mean Rigid Shapes – Application to the 3D case. In *Image Fusion and Shape Variability Techniques (16th Leeds Annual Statistical Workshop)*, pages 178–185, 1996.

[202] S. Pllana and T. Fahringer. Performance prophet: A performance modeling and prediction tool for parallel and distributed programs. *Proc. 2005 International Conference on Parallel Processing ICPP-05, Oslo*, 26(11):509–516, June 2005.

[203] M. Poldner and H. Kuchen. On Implementing the Farm Skeleton. In *Parallel Processing Letters*, pages 117–131, 2008.

[204] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat. Comparing ICP variants on real-world data sets. *Autonomous Robots*, 34(3):133–148, 2013.

[205] J. Pons, R. Keriven, and O. Faugeras. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *International Journal of Computer Vision*, 72(2):179–193, 2007.

[206] H. Pottmann, S. Leopoldseder, and M. Hofer. Simultaneous registration of multiple views of a 3D object. In *Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 265–270, 2002.

[207] H. Pottmann, Q. Huang, Y. Yang, and S. Hu. Geometry and convergence analysis of algorithms for registration of 3D shapes. *Int. J. Computer Vision*, 67(3):277–296, 2006.

[208] M. Previtali, L. Barazzetti, R. Brumana, and M. Scaioni. Laser scan registration using planar features. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 45, 2014.

[209] PrimeSense. PrimeSense. http://www.primesense.com/ Accessed February 2015.

[210] K. Pulli. Multi-view registration for large data sets. *3D Digital Imaging and Modelling*, pages 160–168, 1999.

[211] D. Qiu, S. May, and A. Nuchter. GPU-Accelerated Nearest Neighbor Search for 3D Registration. In *Proceedings of the 7th International Conference on Computer Vision Systems: Computer Vision Systems*, ICVS '09, pages 194–203, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-04666-7. doi: 10.1007/978-3-642-04667-4_20. URL http://dx.doi.org/10.1007/978-3-642-04667-4_20.

[212] H. L. Ramsden, G. Sürmeli, S. G. McDonagh, and M. F. Nolan. Laminar and dorsoventral molecular organization of the medial entorhinal cortex revealed by large-scale anatomical analysis of gene expression. *PLoS computational biology*, 11(1), 2015.

[213] A. Rangarajan, H. Chui, E. Mjolsness, S. Pappu, L. Davachi, P. man Rakic, and J. Duncan. A robust point-matching algorithm for autoradiograph alignment. *Medical Image Analysis*, 1(4):379–398, 1997.

[214] Stuttgart Range Image Database. Department of Image Understanding. http://range.informatik.uni-stuttgart.de/ Accessed October 2012.

[215] A. Rasoulian, R. Rohling, and P. Abolmaesumi. Group-wise registration of point sets for statistical shape models. *IEEE Trans Med Imaging*, 31(11):2025–2034, Nov 2012.

[216] T. Rauber and G. Rünger. *Parallel programming: For multicore and cluster systems*. Springer Science & Business, 2013.

[217] P. A. Revenga, J. Sérot, J. L. Lázaro, and J. P. Derutin. A Beowulf-Class Architecture Proposal for Real-Time Embedded Vision. In *Proceedings of the 17th International Symposium on Parallel and Distributed Processing*, IPDPS '03, pages 232–242, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-1926-1. URL http://dl.acm.org/citation.cfm?id=838237.838308.

[218] M. Rosenblatt. Remarks on Some Nonparametric Estimates of a Density Function. *The Annals of Mathematical Statistics*, 27(3):832–837, 09 1956. doi: 10.1214/aoms/1177728190. URL http://dx.doi.org/10.1214/aoms/1177728190.

[219] G. Roth. Registering two overlapping range images. In *3-D Digital Imaging and Modelling. Proceedings. Second International Conference on*, pages 191–200, 1999. doi: 10.1109/IM.1999.805349.

[220] D. Rueckert and J. Schnabel. Medical Image Registration. In T. M. Deserno, editor, *Biomedical Image Processing*, Biological and Medical Physics, Biomedical Engineering, pages 131–154. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-15815-5. doi: 10.1007/978-3-642-15816-2_5. URL http://dx.doi.org/10.1007/978-3-642-15816-2_5.

[221] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proc. of the third Intl. Conference on 3D Digital Imaging and Modelling*, volume 3, pages 145–152, 2001.

[222] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3D Model Acquisition. *ACM Trans. Graph.*, 21(3):438–446, July 2002. ISSN 0730-0301. doi: 10.1145/566654.566600. URL http://doi.acm.org/10.1145/566654.566600.

[223] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3D Model Acquisition. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH, pages 438–446, New York, NY, USA, 2002. ACM. ISBN 1-58113-521-1. doi: 10.1145/566570.566600. URL http://doi.acm.org/10.1145/566570.566600.

[224] S. R. Sain. *Adaptive kernel density estimation*. PhD thesis, Rice University, Houston, Texas, August 1994.

[225] S. R. Sain and D. W. Scott. Zero-Bias Bandwidths for Locally Adaptive Kernel Density Estimation. *Scandinavian Journal of Statistics*, 29:441–460, 2002.

[226] J. Salvi, C. Matabosch, D. Fofi, and J. Forest. A review of recent range image registration methods with accuracy evaluation. *Image and Vision Computing*, 25(5):578–596, 2007. ISSN 0262-8856. doi: http://dx.doi.org/10.1016/j.imavis.2006.05.012. URL http://www.sciencedirect.com/science/article/pii/S0262885606001594.

[227] Scanalyze. Scanalyze. http://graphics.stanford.edu/software/scanalyze/ Accessed February 2015.

[228] O. Schall, A. Belyaev, and H. Seidel. Robust filtering of noisy scattered point data. In *IEEE / Eurographics Symposium on Point-Based Graphics.*, pages 71–77, 2005.

[229] A. Scheenstra, A. Ruifrok, and R. C. Veltkamp. A survey of 3D face recognition methods. In *Audio-and Video-Based Biometric Person Authentication*, pages 891–899. Springer, 2005.

[230] D. W. Scott and S. R. Sain. Multidimensional Density Estimation. In C. R. Rao, E. J. Wegman, and J. L. Solka, editors, *Data Mining and Data Visualization*, volume 24 of *Handbook of Statistics*, pages 229–261. Elsevier, 2005. doi: http://dx.doi.org/10.1016/S0169-7161(04)24009-3. URL http://www.sciencedirect.com/science/article/pii/S0169716104240093.

[231] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 519–528, June 2006. doi: 10.1109/CVPR.2006.19.

[232] J. K. Seo, G. C. Sharp, and S. W. Lee. Range data registration using photometric features. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 1140–1145. IEEE, 2005.

[233] R. Shams, P. Sadeghi, R. Kennedy, and R. Hartley. Parallel computation of mutual information on the gpu with application to real-time registration of 3D

medical images. *Computer methods and programs in biomedicine*, 99(2):133–146, 2010.

[234] G. C. Sharp, S. W. Lee, and D. K. Wehe. Toward multiview registration in frame space. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 4, pages 3542–3547 vol.4, 2001. doi: 10.1109/ROBOT.2001.933166.

[235] G. C. Sharp, S. W. Lee, and D. K. Wehe. Multiview Registration of 3D Scenes by Minimizing Error Between Coordinate Frames. In *Proceedings of the 7th European Conference on Computer Vision*, ECCV '02, pages 587–597, London, UK, 2002. Springer-Verlag. ISBN 3-540-43744-4. URL http://dl.acm.org/citation.cfm?id=645316.649200.

[236] G. C. Sharp, S. W. Lee, and D. K. Wehe. Multiview Registration of 3D Scenes by Minimizing Error between Coordinate Frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1037–1050, 2004. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2004.49.

[237] S. J. Sheather. Density Estimation. *Statistical Science*, 19(4):588–597, 11 2004.

[238] L. Silva, O. R. P. Bellon, and K. L. Boyer. Enhanced, robust genetic algorithms for multiview range image registration. In *3DIM03*, pages 268–275, 2003.

[239] L. M. Silva, B. Veer, and J. G. Silva. How to get a fault-tolerant farm. In *World Transputer Congress*, pages 923–938, Aachen, Germany, September 1993.

[240] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London, 1986.

[241] J. S. Simonoff. *Smoothing methods in statistics*. Springer, New York, 1996.

[242] D. B. Skillicorn, J. Hill, and W. F. McColl. Questions and Answers about BSP. *Scientific Programming*, 6:249–274, January 1997.

[243] Softkinetic. Softkinetic. http://www.softkinetic.com/ Accessed February 2015.

[244] R. Song, Y. Liu, R. R. Martin, and P. L. Rosin. Saliency-guided integration of multiple scans. In *Computer Vision and Pattern Recognition (CVPR), 2012*

IEEE Conference on, pages 1474–1481, June 2012. doi: 10.1109/CVPR.2012.6247836.

[245] R. Song, Y. Liu, Y.. Zhao, R. Martin, and P. Rosin. An Evaluation Method for Multiview Surface Reconstruction Algorithms. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 387–394, Oct 2012. doi: 10.1109/3DIMPVT.2012.24.

[246] D. P. Spooner, S. A. Jarvis, J. Cao, S. Saini, and G. R. Nudd. Local grid scheduling techniques using performance prediction. *IEE Proceedings - Computers and Digital Techniques*, 150:87–96(9), March 2003. URL http://digital-library.theiet.org/content/journals/10.1049/ip-cdt_20030280.

[247] F. Stein and G. Medioni. Structural indexing: Efficient 3D object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):125–145, 1992.

[248] C. V. Stewart. Robust parameter estimation in computer vision. *SIAM Reviews*, 41:513–537, 1999.

[249] A. J. Stoddart and A. Hilton. Registration of multiple point sets. In *Proc. 13th Int. Conference on Pattern Recognition*, volume 2, pages 40–44, 1996.

[250] J. Stowers, M. Hayes, and A. Bainbridge-Smith. Altitude control of a quadrotor helicopter using depth map from microsoft kinect sensor. In *Mechatronics (ICM), 2011 IEEE International Conference on*, pages 358–362, April 2011. doi: 10.1109/ICMECH.2011.5971311.

[251] R. Szeliski. *Computer vision: algorithms and applications*. Springer, 2010.

[252] G. K. L. Tam, Z. Cheng, Y. Lai, F. C. Langbein, Y. Liu, D. Marshall, R. R. Martin, X. Sun, and P. L. Rosin. Registration of 3D Point Clouds and Meshes: A Survey From Rigid to Non-Rigid. *Visualization and Computer Graphics, IEEE Transactions on*, 19(7):1199–1217, 2013.

[253] T. Tamaki, M. Abe, B. Raytchev, and K. Kaneda. Softassign and EM-ICP on GPU. In *Networking and Computing (ICNC), 2010 First International Conference on*, pages 179–183. IEEE, 2010.

[254] J. W. H. Tangelder and R. C. Veltkamp. A survey of content based 3D shape retrieval methods. *Multimedia tools and applications*, 39(3):441–471, 2008.

[255] M. Taron, N. Paragios, and M. P. Jolly. Modelling Shapes with Uncertainties: Higher Order Polynomials, Variable Bandwidth Kernels and Non-Parametric Density Estimation. In *IEEE International Conference in Computer Vision (ICCV)*, 2005.

[256] G. R. Terrell and D. W. Scott. Variable Kernel Density Estimation. *The Annals of Statistics*, 20(3):1236–1265, 09 1992. doi: 10.1214/aos/1176348768. URL http://dx.doi.org/10.1214/aos/1176348768.

[257] D. Thain, T. Tannenbaum, and M. Livny. Distributed computing in practice: the Condor experience. *Concurrency and Computation: Practice and Experience*, 17(2-4):323–356, 2005. ISSN 1532-0634. doi: 10.1002/cpe.938. URL http://dx.doi.org/10.1002/cpe.938.

[258] The Stanford 3D Scanning Repository. The Stanford 3D Scanning Repository. http://graphics.stanford.edu/data/3Dscanrep/ Accessed February 2015.

[259] D. Thomas. *Range Image Registration Based on Photometry*. PhD thesis, The National Institute of Informatics, SOKENDAI, Tokyo, Japan., July 2012. http://researchmap.jp/diegothomas/.

[260] D. Thomas and Y. Matsushita. Robust Simultaneous 3D Registration via Rank Minimization. In *3D Im. Modelling Proc. Vis. and Trans 2012 (3DIMPVT).*, pages 73–80, 2012.

[261] D. Thomas and A. Sugimoto. A flexible scene representation for 3D reconstruction using an RGB-D camera. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2800–2807. IEEE, 2013.

[262] R. Toldo, A. Beinat, and F. Crosilla. Global registration of multiple point clouds embedding the Generalized Procrustes Analysis into an ICP framework. In *3DPVT – 3D Data Processing Visualisation Transmission*, 2010.

[263] A. Torsello, E. Rodola, and A. Albarelli. Multi-view registration via graph diffusion of dual quaternions. In *IEEE Conference Computer Vision and Pattern Recognition*, pages 2441–2448, 2011.

[264] E. Trucco and A. Verri. *Introductory techniques for 3D computer vision*, volume 201. Prentice Hall Englewood Cliffs, 1998.

[265] Y. Tsin and T. Kanade. A correlation-based approach to robust point set registration. In *Computer Vision ECCV 2004*, pages 558–569. Springer, 2004.

[266] A. Turlach. Bandwidth selection in kernel density estimation: A review. *CORE and Institut de Statistique*, 1993.

[267] G. L. Valiant. A bridging model for parallel computation. *Commun. ACM*, 33 (8):103–111, Aug 1990. ISSN 0001-0782. doi: 10.1145/79173.79181. URL http://doi.acm.org/10.1145/79173.79181.

[268] P. A. van den Elsen, E. J. D. Pol, and M. A. Viergever. Medical image matching-a review with classification. *Engineering in Medicine and Biology Magazine, IEEE*, 12(1):26–39, Mar 1993. ISSN 0739-5175. doi: 10.1109/51.195938.

[269] R. Vinesh and F. Kiran. *Reverse Engineering: An Industrial Perspective*. Springer, 2007.

[270] G. Vogiatzis, P. Torr, S. M. Seitz, and R. Cipolla. Reconstructing relief surfaces. In *British Machine Vision Conference*. Citeseer, 2004.

[271] A. Vrubel, O. R. P. Bellon, and L. Silva. A 3D reconstruction pipeline for digital preservation. In *Computer Vision and Pattern Recognition 2009 IEEE Computer Society Conference on*, pages 2687–2694, 2009.

[272] M. P. Wand and M. C. Jones. Comparison of smoothing parameterizations in bivariate kernel density estimation. *Journal of the American Statistical Association*, 88(422):520–528, 1993. doi: 10.1080/01621459.1993.10476303.

[273] M. P. Wand and M. C. Jones. Multivariate plug-in bandwidth selection. *Computational Statistics*, 9(2):97–116, 1994.

[274] F. Wang, B. C. Vemuri, and A. Rangarajan. Groupwise point pattern registration using a novel CDF-based Jensen-Shannon Divergence. In *Computer Vision and Pattern Recognition, 2006. CVPR 2006. Proceedings of the 2006 IEEE Computer Society Conference on*, volume 1, pages 1283–1288, June 2006. doi: 10.1109/CVPR.2006.131.

[275] M. H. Wegkamp. Quasi-universal bandwidth selection for kernel density estimators. *Canadian Journal of Statistics*, 27(2):409–420, 1999. ISSN 1708-945X. doi: 10.2307/3315649. URL http://dx.doi.org/10.2307/3315649.

[276] W. M. Wells. Statistical approaches to feature-based object recognition. *International Journal of Computer Vision*, 21(1-2):63–98, 1997.

[277] J. West, J. M. Fitzpatrick, M. Y. Wang, B. M. Dawant, C. R. Maurer Jr., R. M. Kessler, and R. J. Maciunas. Retrospective intermodality registration techniques for images of the head: surface-based versus volume-based. *Medical Imaging, IEEE Transactions on*, 18(2):144–150, Feb 1999. ISSN 0278-0062. doi: 10.1109/42.759119.

[278] M. Whitty, S. Cossell, K. Dang, J. Guivant, and J. Katupitiya. Autonomous navigation using a real-time 3D point cloud. In *Australasian Conference on Robotics and Automation*, 2010.

[279] J. Williams and M. Bennamoun. Simultaneous registration of multiple corresponding point sets. *Computer Vision and Image Understanding*, 81(1):117–142, 2001.

[280] J. Xiao, J. Chai, and T. Kanade. A Closed-Form Solution to Non-Rigid Shape and Motion Recovery. In *The 8th European Conference on Computer Vision (ECCV 2004)*, May 2004.

[281] X. Yangand, H. Qiao, and Z. Liu. Outlier robust point correspondence based on gnccp. *Pattern Recognition Letters*, 2015.

[282] A. Yarkhan, K. Seymour, K. Sagi, Z. Shi, and J. Dongarra. Recent Developments in Gridsolve. *IJHPCA*, 20(1):131–141, 2006.

[283] Y. Yemez and C. J. Wetherilt. A volumetric fusion technique for surface reconstruction from silhouettes and range data. *Computer Vision and Image Understanding*, 105(1):30–41, 2007.

[284] D. Zhang and M. Hebert. Harmonic maps and their applications in surface matching. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on. (CVPR).*, volume 2, pages 524–530, 1999. doi: 10.1109/CVPR.1999.784731.

[285] Z. Zhang. Microsoft Kinect Sensor and Its Effect. *MultiMedia, IEEE*, 19(2):4–10, Feb 2012. ISSN 1070-986X. doi: 10.1109/MMUL.2012.24.

[286] Y. Zheng, J. Jestes, J. M. Phillips, and F. Li. Quality and efficiency for kernel density estimates in large data. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 433–444. ACM, 2013.

[287] H. Zhou and Y. Liu. Accurate integration of multi-view range images using k-means clustering. *Pattern Recognition*, 41(1):152–175, 2008.

[288] H. Zhou, Y. Liu, L. Li, and B. Wei. A clustering approach to free form surface reconstruction from multi-view range images. *Image and Vision Computing*, 27 (6):725–747, 2009.

[289] Q. Zhou and V. Koltun. Dense Scene Reconstruction with Points of Interest. *ACM Trans. Graph.*, 32(4):112:1–112:8, July 2013. ISSN 0730-0301. doi: 10.1145/2461912.2461919. URL http://doi.acm.org/10.1145/2461912.2461919.

[290] B. Zitova and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, 2003. ISSN 0262-8856. doi: http://dx.doi.org/10.1016/S0262-8856(03)00137-9. URL http://www.sciencedirect.com/science/article/pii/S0262885603001379.

[291] M. Zwicker, H. Pfister, J. van Baar, and M. Gross. Surface splatting. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 371–378, New York, NY, USA, 2001. ACM. ISBN 1-58113-374-X. doi: 10.1145/383259.383300. URL http://doi.acm.org/10.1145/383259.383300.