Recognizing Three-Dimensional Objects using Parameterized Volumetric Models

Díbio Leandro Borges

Ph.D.
University of Edinburgh
1996

Abstract

This thesis addresses the problem of recognizing 3-D objects, using shape information extracted from range images, and parameterized volumetric models. The domain of the geometric shapes explored is that of complex curved objects with articulated parts, and a great deal of similarity between some of the parts. These objects are exemplified by animal shapes, however the general characteristics and complexity of these shapes are present in a wide range of other natural and man-made 3-D objects.

In model-based object recognition three main issues constrain the design of a complete solution: representation, feature extraction, and interpretation. This thesis develops an integrated approach that addresses these three issues in the context of the above mentioned domain of objects. For representation I propose a composite description using globally deformable superquadrics and a set of volumetric primitives called geons: this description is shown to have representational and discriminative properties suitable for recognition. Feature extraction comprises a segmentation process which develops a method to extract a parts-based description of the objects as assemblies of deformable superquadrics. Discontinuity points detected from the images are linked using an "active contour" minimization technique, and deformable superquadric models are fitted to the resulting regions afterwards. Interpretation is split into three components: classification of parts, matching, and pose estimation. A Radial Basis Function [RBF] classifier algorithm is presented in order to classify the superquadrics shapes derived from the segmentation into one of twelve geon classes. The matching component is decomposed into two stages: first, an indexing scheme which makes effective use of the output of the [RBF] classifier in order to direct the search to the models which contain the parts identified. This makes the search more efficient, and with a model library that is organized in a meaningful and robust way, permits growth without compromising performance. Second, a method is proposed where the hypotheses picked from the index are searched using an Interpretation Tree algorithm combined with a quality measure to evaluate the bindings and the final valid hypotheses based on Possibility Theory, or Theory of Fuzzy Sets. The valid hypotheses ranked by the matching process are then passed to the pose estimation module. This module uses a Kalman Filter technique that includes the constraints on the articulations as perfect measurements, and as such provides a robust and generic way to estimate pose in object domains such as the one approached here.

These techniques are then combined to produce an integrated approach to the object recognition task. The thesis develops such an integrated approach, and evaluates its performance in the sample domain. Future extensions of each technique and the overall integration strategy are discussed.

Acknowledgements

First, and foremost, I would like to thank my supervisor, Dr. Robert Fisher, for all the guidance, support and encouragement. My sincere thanks also to Dr. Manuel Trucco and to Dr. John Hallam for functioning as my second supervisors and for sharing their knowledge with me.

I feel fortunate to have had the chance to work in such an stimulating research environment as the A.I. Department in Edinburgh. Although taking the risk of forgetting someone I would like to mention Edvaldo Bispo, David Eggert, Philippe Fillatreau, Andrew Fitzgibbon, Bridget Hallam, Taehee Kim, Mark Orr, Giovanni Pettinaro, Maurizio Pilu, Martin Waite, Ashley Walker, David Wren, Mark Wright, and express my gratitude for many useful comments, constructive criticisms, and for the many meals and pub visits we enjoyed together.

Before coming to Edinburgh I did my Master's degree in the Computer Science Department, University of Brasilia, where I was introduced into Artificial Intelligence and Computer Vision. I appreciate the support and the friendship kept throughout these years of Dr. Gentil Lucena.

Life would not have been the same without the friendship and support of many people in these past three years in Edinburgh. Also, from Brazil, my brother Lino, my sister Liliana, and my mother have provided constant support through innumerable phone calls and letters. My special thanks to all of you.

I would like to thank also the National Council for Research and Scientific Development (CNPq), Ministry of Science and Technology, Brazil, for the financial support of this work. In particular, my thanks to Nelson Prugner for providing all the assistance when I needed it.

Declaration

I hereby declare that I composed this thesis entirely myself and that it describes my own research.

Díbio Leandro Borges
Edinburgh
June 1st, 1996

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recognition of 3-D objects is an important problem that an intelligent agent has to solve for reasoning about the outside world. Developing an understanding of object recognition is one of the principal goals in the field of Artificial Intelligence known as Computer Vision.

We perform this task millions of times throughout our lives with such great accuracy and speed that it is often taken for granted. However, computer recognition of 3-D objects has remained an area where the proposed methods work only for a small set of objects mostly composed of planar faces. The few attempts to work with more complex curved objects have been restricted to small subsets of particular objects, such as man-made utensils.

This thesis is concerned with 3-D object recognition, *i.e.* identification and location, of complex curved objects with articulations. A class of natural forms is chosen to constrain the problem, thus making it possible to separate and address the issues involved. As an example of what the research has achieved, we can recognize, from range data, similar animal shapes like cows and horses, and be sufficiently accurate to discriminate between them.

1.1 Model-Based 3-D Object Recognition

In the model-based vision paradigm, recognition of 3-D objects comprises the identification of an object present in the scene and determining its location relative to sensor position information. In order to identify what the object is, the system must have *a priori* knowledge about objects, and this knowledge is provided by building a structured geometric model in a local reference frame emphasizing particular characteristics of each object. The companion problem is to find where the object is in the scene, *i.e.* a transformation in 3-D space mapping the local reference frame of the model to the data.

Three major elements can be distinguished in this paradigm: representation, feature extraction and interpretation. This thesis deals with these components using range images of the 3-D objects as the raw input.

1.1.1 Range Images

Recognition of 3-D objects often requires depth information about the scene being analyzed. Intensity images are common inputs in Computer Vision, and depth information is usually extracted from these images by "shape-from-X" techniques, such as shape from shading, shape from stereo, shape from focus, and shape from motion. However, depth information extracted from intensity images is still sparse, of low quality (*i.e.* with a great deal of noise), and often constrained to unrealistic assumptions about the light and the surface of the objects.

Range images are images where each pixel represents a quantized measure of the depth of the object surface relative to the viewer. Range sensors are becoming widely available with increasing developments in technology. Depth information is directly available, which makes range images more suitable for the task of 3-D object recognition.

One of the most popular range sensing techniques is triangulation. In this technique a laser stripe is projected onto the scene, and one or two cameras are used to determine the location of the pattern on the surface of the object. In our laboratory a triangulation based range sensor can provide range images with an accuracy of 0.15mm. All the range images used in this thesis were acquired using this sensor. Figures 1.1 and 1.2 show a photograph of the Edinburgh Laser Striper and a diagram of its main features.

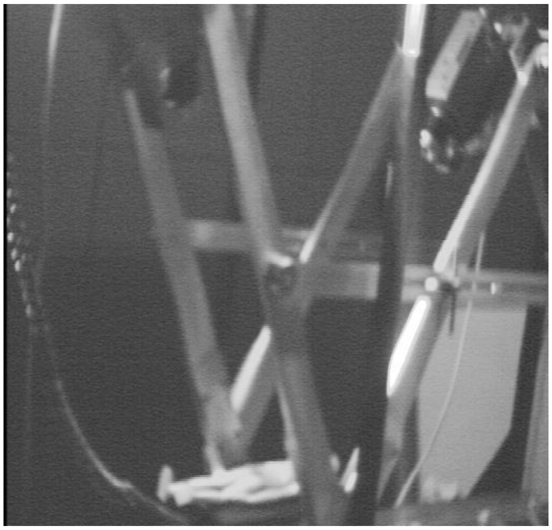


Figure 1.1: Photograph of the Edinburgh Laser Striper which was used to acquire the range images in this thesis.

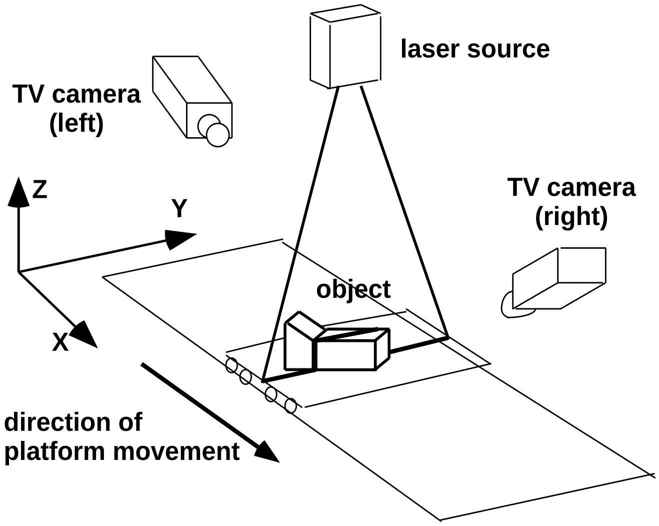
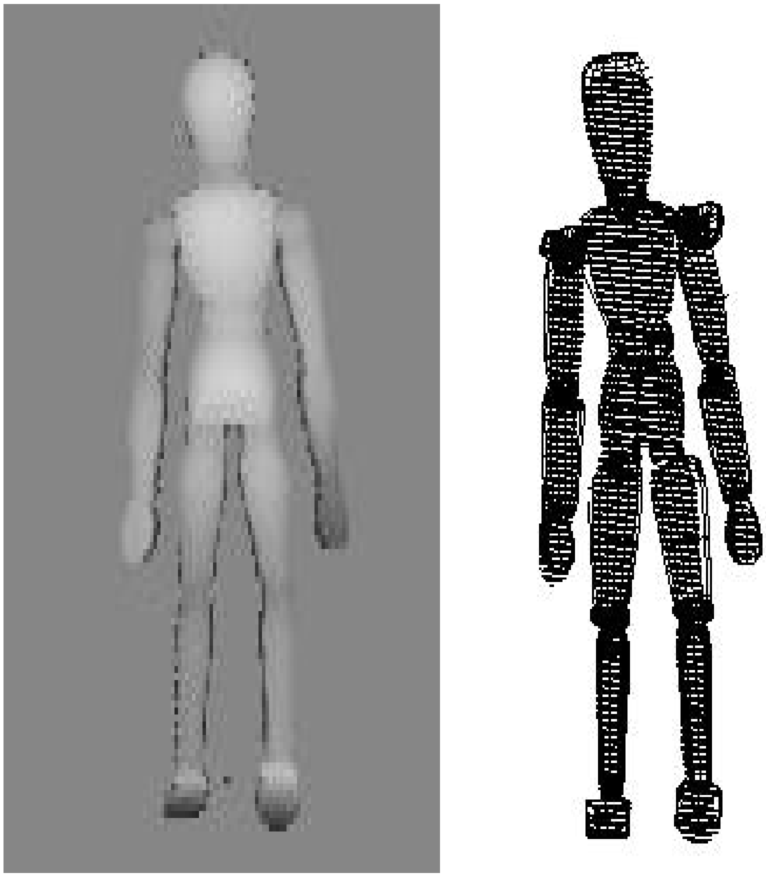


Figure 1.2: Schematic diagram of the Edinburgh Laser Striper.

1.2 The Problem

The goal of this research is to recognize 3-D complex curved objects in range images in terms of volumetric primitives which describe parts of the object. Recognition here comprises both identification and localization. Knowledge about the objects to be recognized is organized in a model library, where a model containing geometric information about each part and relationships between them is given for each object. One typical example showing respectively an input and the final result of the algorithms and techniques described here can be seen in Figures 1.3 (a) and 1.3 (b). Section 1.3 summarizes the approach developed to achieve this.

The domain of the geometric shapes explored is one of complex curved objects with articulated parts, with a great deal of similarity between some of the parts. These objects are exemplified by animal shapes, however the general characteristics and complexity of these shapes are present in a wide range of other natural and man-made 3-D objects.



(a) (b)

Figure 1.3: (a) Range image of an articulated object; (b) Recognized model (in wireframe) overlayed on 3-D data of object in (a) indicating successful identification and location. This a typical example of object recognition achieved in this thesis.

1.2.1 Important Issues

Recognition is a high level task which involves finding the best interpretation of a scene according to some knowledge provided to the system. In approaching 3-D object recognition in the scenario identified in this work the important issues that arise are:

- **How to represent 3-D objects?**

  Shape representation has a purpose, namely to allow a process of recognition to take place. This decision constrains the design of the recognition strategy, and it makes explicit the physical forms favoured by the description process. This issue also affects the approach's efficiency, since the primitives used to describe the object constrain the complexities of the model, the size of the search space in matching, its reliance on verification, and the ease of recovery of the primitives from the 3-D image data.

- **What features can be extracted to give evidence for identification and location of 3-D objects?**

  This issue relates to the segmentation problem, in which the interpretation of the scene begins by finding regions of interest separating the object from irrelevant data, and then estimating values of features. These features will be used directly or indirectly as evidence in making hypotheses about the objects.

- **How to match, or how to organize the search process of comparing the features extracted from the scene to the models in the database?**

  Recognition is not a yes/no problem, and it has to be efficient in searching the models. A degree of certainty about the pairings needs to be determined since feature correspondences are not always exact and some objects are similar. The search has to effectively use the knowledge about the features extracted to narrow down the number of models in the database that will be checked.

- **How to locate the best hypotheses found in the original 3-D data?**

  This issue is related to the verification of the hypotheses. The models in the library are parameterized geometric descriptions of 3-D objects. They represent a class of objects having similar geometric descriptions, and since parts can be articulated or not be found in the 3-D data due to occlusion, computing a transformation that brings the matched model over the 3-D data is an important part of the recognition process.

1.3 The Approach

In model-based object recognition three main elements constrain the design of a complete solution: representation, feature extraction, and interpretation. This thesis develops an integrated approach that provides answers to each of these three elements in the context of the chosen domain of objects.

For representation I propose a composite description using globally deformable superquadrics and a set of volumetric primitives called geons: this description is shown to have representational and discriminative properties suitable for recognition. Feature extraction comprises a segmentation process which develops a method to extract a parts-based description of the objects as assemblies of deformable superquadrics. Discontinuity points detected from the images are linked using an "active contour" minimization technique, and deformable superquadric models are fitted to the resulting regions afterwards. Interpretation is split into three components: classification of parts, matching, and pose estimation. A Radial Basis Function [RBF] classifier algorithm is presented in order to classify the superquadric shapes derived from the segmentation into one of twelve geon classes. The matching component is decomposed into two stages. Firstly, an indexing scheme makes effective use of the output of the [RBF] classifier in order to direct the search to the models which contain the parts identified. This makes the search more efficient, and with a model library that is organized in a meaningful and robust way, permits growth without compromising performance. Secondly, a method is proposed where the hypotheses identified by the indexing scheme are searched using an Interpretation Tree algorithm combined with a quality measure to evaluate the bindings and the final valid hypotheses based on Possibility Theory, or the Theory of Fuzzy Sets. The valid hypotheses ranked by the matching process are then passed to the pose estimation module. This module uses a Kalman Filter technique that includes the constraints on the articulations as perfect measurements, and as such provides a robust and generic way to estimate pose in object domains such as the one approached here.

Figure 1.4 shows a functional diagram of the recognition approach presented in this thesis.

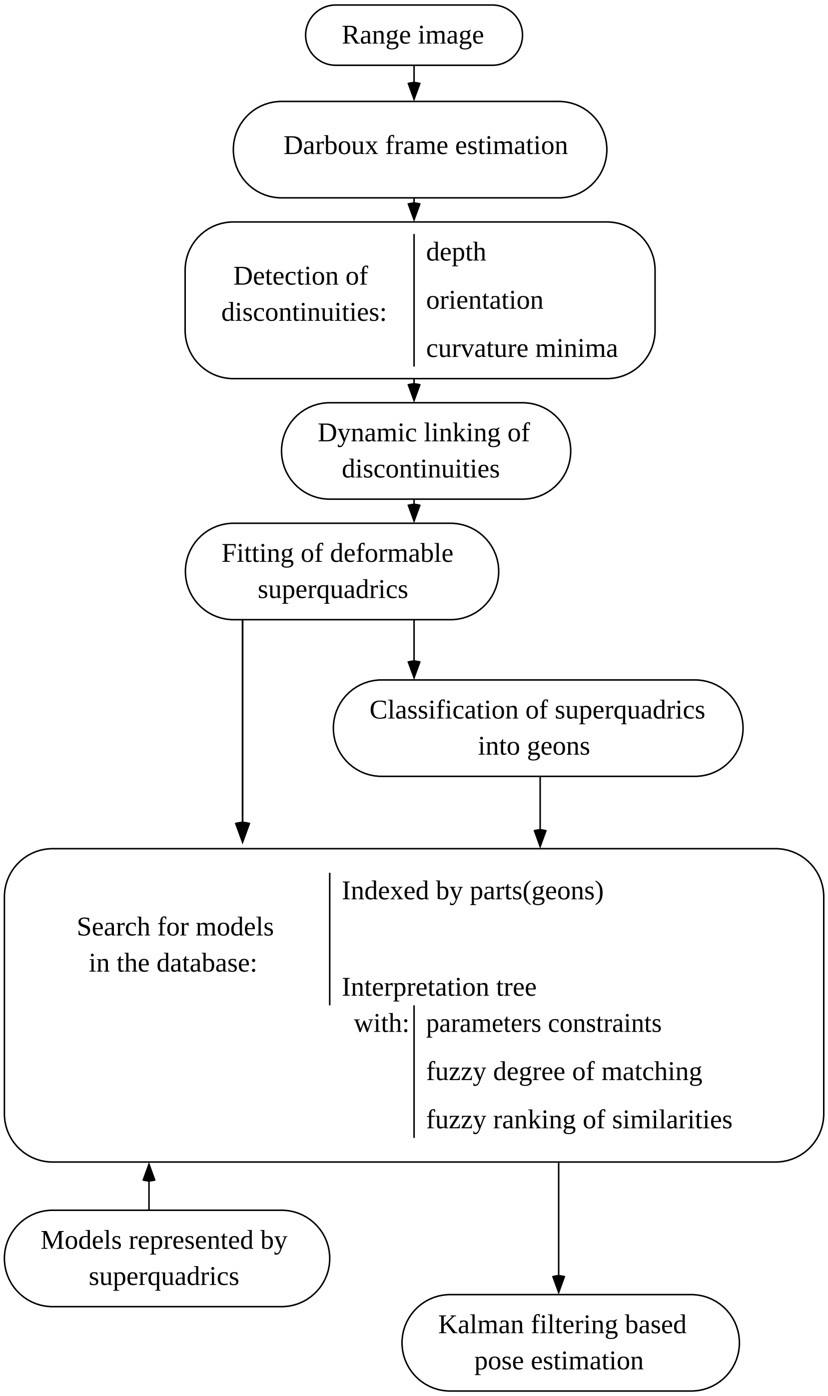


Figure 1.4: Functional diagram of the recognition approach developed in this thesis.

1.4 Organization of this Thesis

Chapter 2 discusses previous work done in model-based object recognition. The different approaches are organized according to the representation primitives they used: points and lines, surfaces, and volumes. Characteristics of feature extraction, modelling, and matching mechanisms are considered.

Chapter 3 presents an integrated approach to segmenting 3-D articulated objects into prototypical parts. Discontinuities on the object surface are detected and then linked utilizing an active contour driven by information about the principal directions on the points. Deformable superquadrics are then fitted to each region found providing volumetric features to be classified and passed to the matching process.

Chapter 4 discusses the use of quantitative and qualitative features in a shape representation for the purpose of 3-D object recognition. It presents such a representation, and it develops a multi-dimensional classifier algorithm to map the superquadric parameters estimated during the segmentation into a set of twelve pre-defined volumetric primitives.

Chapter 5 presents a matching approach that builds the correspondences between data and model in two stages. First, with the class identities for the object's parts it searches the model library for the models indexed under those class labels. Second, these model hypotheses are ordered and searched through an interpretation tree for feature correspondences. The geometric constraints are evaluated by a fuzzy predicate, and the valid hypotheses are then ranked according to a quality measure of similarity. The highest ranked valid hypotheses are passed for verification to the pose estimation module.

Chapter 6 shows that pose estimation and constraint satisfaction for localizing and verifying the hypotheses matched can be realized in an integrated formulation. The tool used for this integration is the Kalman filter. Formulation of an Iterated Extended Kalman Filter fuser is presented for this problem, together with results and consideration of related work.

Chapter 7 summarizes the approach presented in this thesis and recommends future directions for research.

1.5 Contributions

This thesis explores a number of topics, the major contributions reported here being:

- A model-based recognition approach for complex 3-D objects using parameterized volumetric parts-based models is presented. Identification and location are achieved; which is the first attempt in this domain.
- A procedure is presented which combines several techniques to provide a solution to parts-based segmentation of 3-D objects from range data. The approach relies on computing stable estimates of the differential structure of the object's surfaces, followed by a dynamic grouping of the detected surface discontinuities and a fitting of deformable superquadric models to the segmented regions.
- A shape representation for parts of 3-D objects is introduced which combines two volumetric primitives, deformable superquadrics and geons, in order to explore quantitative and qualitative properties to provide an efficient representation for the recognition task.
- A method which enables the system to classify 3-D parts represented by deformable superquadrics into a subset of geon primitives is presented. Results are analyzed and compared with related work in classification of similar features.
- An indexing scheme is developed which uses the subset of geon primitives as symbolic keys. Models are organized within the library into groups according to the presence of a salient (biggest part) geon primitive. The output of the classifier, which precedes the matching, is used to order and make the initial hypotheses to be searched within the models library. This indexing scheme makes effective use of the available information and narrows the search space for further matching.
- A constrained search algorithm is developed for parts-based parameterized objects which extends the Interpretation Tree approach using novel constraints and fuzzy predicates for evaluation.

- A quality measure of similarity is presented which ranks hypotheses searched through the constraint based matching. The measure is developed using Fuzzy Sets to describe the feature pairings followed by a global evaluation integrating the outputs of the fuzzy predicates.

- Location of the best hypotheses is achieved by a pose estimation algorithm based on Kalman Filtering. A method for pose estimation of individual deformable superquadric parts is shown, and is used as the initial estimates for computing the pose of articulated objects in a second stage. Both stages are developed in a Kalman Filtering framework. The algorithm shows how to include constraints in the estimation process, and it achieves a final pose integrating all the information.

Chapter 2

Approaches to 3-D Object Recognition

Since the seminal work of [Roberts 65] many different frameworks have been proposed to fully achieve 3-D object recognition. Comprehensive reviews until 1986 can be found in [Binford 82] on model-based vision systems, in [Besl & Jain 85] on requirements and existing approaches to 3-D recognition, and [Chin & Dyer 86] with a thorough review of 2-D and 3-D model-based systems. For more recent reviews dealing with research done after 1986 one should look at [Brady *et al.* 89] and [Arman & Aggarwal 93] which focus on systems and issues using dense range images.

The main issues to be considered for approaching 3-D object recognition are:

- Feature extraction; What and how features should be extracted from the sensed image.
- Model representation; How to represent the models to be recognized in a coherent and efficient way making use of the distinctive features of the object.
- Matching strategy; How to establish the correspondences between the features extracted from the scene and the models in the database.

These issues can not be dealt with separately though, they all depend on each other in order for the task of recognition to be achieved. However, the central issue and decisive one is the Model Representation. Once a representation has been chosen that is well suited for the classes of objects to be recognized, the feature extraction and the matching modules can be designed.

A model representation for 3-D object recognition ought to have properties suitable for extracting and matching purposes, and in general the following criteria have been proposed to judge them [Marr 82]:

- Accessibility
- Scope and uniqueness
- Sensitivity and stability

In the following sections, the main approaches to 3-D object recognition are classified according to the geometric primitives they use to represent the classes of objects they aim to identify.

2.1 Approaches using Points and Lines as Modelling Primitives

Points and lines are the simplest features to extract from images, and a good deal of 3-D structure of a scene can be inferred from them. The richness of information one can infer from a line drawing promptly shows that. However, for model-based vision purposes the main difficulty in using points and lines as modelling primitives resides on the fact that models constructed with them are high in complexity, therefore causing the search to be complex as well, and reliance on verification to be very strong. Table 2.1 summarizes the main approaches using these primitives.

[Lowe 85] described the SCERPO system which received, as input, intensity images of known 3-D objects. The system did not require the derivation of depth or surface orientation prior to recognition. Its approach is based on a perceptual organization method that groups the points and lines of the image into structures, and calculates their probability of accidentalness. In the model making process a list of perceptual groupings is included together with the description of the object using point and line primitives. A quantitative mathematical technique is developed to solve exactly for viewpoint and variable object parameters. This technique is a powerful tool for the verification of the matches, and for exploring other matches once the viewpoint is found. Recently, [Lowe 91] has extended his viewpoint technique to locate curved objects in intensity images.

The TINA system [Pollard *et al.* 87] combined two or more three-space descriptions obtained from edge-based binocular stereo to recognize a 3-D object. The matching was done first by choosing a set of edge segments on which to focus the search, and then using pairwise geometric constraints on the selected set until maximum correspondence was achieved. Estimating the pose of the object followed by applying least-squares for the best rigid rotation and translation transformations as in the technique proposed by [Faugeras & Hebert 86].

[Reid & Brady 93] proposed an algorithm to recognize classes of objects from range data. The aim was to recognize pallets in an industrial environment considering some variability in shape and scale of the objects. Their system worked with polyhedral objects represented by 3-D points and lines, and included free parameters to account for the different objects in a class. Matching was organized by extending the Interpretation Tree algorithm [Grimson 88] to cope with parameterizations in 3-D models that include modelling the constraints in intervals and checking their consistency by a SUP-INF method similar to [Brooks 81] and [Fisher & Orr 91].

2.2 Approaches using Surfaces as Modelling Primitives

Most of the approaches proposed for 3-D object recognition use surfaces as their modelling primitives, especially because derivation of surface descriptions from images is believed to be simpler than extracting volumetric ones, and it has greater scope than just using points and lines. Table 2.2 shows all the main surface-based approaches with their main characteristics.

[Oshima & Shirai 83] presented an approach for recognition of stacked objects with planar and curved surfaces. The approach is divided into two phases: a learning phase, where the object was shown, and a description of it was built as a relational feature graph with surface type and adjacency information; and a recognition phase where matching takes place by first selecting a "kernel" of reliable regions in the scene

Author	**Sensing Modality**	**Model Representation**	**Matching Strategy**
[Lowe 85]	intensity	points and lines	pose error minimization by optimization
[Pollard *et al.* 87]	binocular stereo	points and lines	search (selected features to focus and rigidity constraints checking)
[Reid & Brady 93]	range	points and lines	search (geometric constraints and continuous-valued constraint propagation)

Table 2.1: 3-D Object Recognition Approaches using Points and Lines as primitives

and then sequentially picking models that include the “kernel”. The match then grows in a model-driven way searching for regions around the “kernel” that correspond to the hypothesized ones from the model. Precise location through pose estimation is not addressed. The system relies on its learning phase for acquiring good descriptions of the scene.

[Bhanu 84] proposed a stochastic labeling technique to perform recognition of 3-D shapes that were first approximated by planar faces. Polygons are used for the approximation and from them features like perimeter, length of radius vectors, and area are extracted for matching. The representation of the model by the various views is linked to an object-centered coordinate system which allows the model to be computed automatically from the input data. Following the extraction of the features from one view of the data a set of initial face-labeling probabilities was computed for matchings between each model face and each face in the sensed view. These probability values are then reevaluated after computing a transformation between the model face and the sensed face, which is applied to other pairs of faces (model-data) to check for the errors in the features. If these errors are large, the probability of the match is lowered. The approach was limited to objects having large, and easy to detect, planar surfaces.

[Grimson & Lozano-Perez 84] modelled objects as polyhedra and presented the Interpretation Tree algorithm to achieve recognition. Surface normals and 3-D points are the main features extracted from the data. Geometric measures such as the distance between faces, angles between 3-D points, and angles between the surface normals are proposed as constraints to prune the search space of the paired features which form the hypotheses. The hypotheses that survive are verified by estimating the transformation that relates the sensed points to the corresponding surface of the object.

[Faugeras & Hebert 86] presented an algorithm to recognize and locate 3-D rigid objects. The objects are modelled by quadrics (points, lines, and planes), and they acquire range data using a laser triangulation platform. The features are extracted from the image by first detecting surface discontinuities, estimating the normals, and then using a region growing strategy to fit the planes. They perform matching between the models and the extracted features by a depth-first search guided by rigidity constraints, calculating a rigid transformation between paired features by their quaternion least-squares technique before attempting another pairing. If the transformation is found, other pairings are then hypothesized and verified in the same way. The main advantage of this approach is that the rigidity constraints are view independent, therefore providing a strong measure for verifying the hypothesis.

[Bolles & Horaud 86] described the 3DPO vision system which recognizes 3-D objects from range images. Models are represented using the following components: an extended CAD model, a feature classification network, a model of planar patches, and a wire-frame model. The feature classification network organizes the features by type and size. Their matching strategy works as a local-feature-focus method for 3-D by selecting and locating a key feature (in this case an edge feature), and then expanding the match by adding the features one by one until location of the object can be computed. The features they concentrated on extracting were edges as straight tangential, straight dihedral, and circular dihedral surface intersections. An off-line procedure determines beforehand the order in which the features are to be selected. Although the 3DPO matching strategy is applicable to other scenarios (*i.e.* other objects), the 3DPO system is restricted to objects with circular arcs and straight dihedral edges.

An evidence-based recognition technique was proposed by [Jain & Hoffman 88]. It is a feature based technique where the evidence is arranged as a rule-base consisting of a set of salient or evidence conditions with corresponding weights for various objects in the database. Attributed surface and line features are used to represent the objects. Three qualitative types of surface patches are used: planar, convex, and concave. The attributes are morphological, patch types, and information about the boundaries. Features are extracted and classified from the range data into the surface patch types by statistical tests. To perform recognition a measure of similarity is computed between the set of extracted features and the model and compared to reach a decision using a parallel certainty inference technique common in expert systems [Luger & Stubblefield 92] (*e.g.* MYCIN [Buchanan & Shortliffe 84]).

[Fisher 89] developed a 3-D object recognition system called IMAGINE (IMAGINE I and II) to identify and locate 3-D curved objects from range data. IMAGINE II analyzes segmented surface patches extracted from range images. The connected surfaces are grouped to form clusters that represent one object; this process is repeated by connecting the clusters so that a tree of clusters with surfaces as leaf nodes is the final input to be exploited in recognition. IMAGINE II is built around the SMS modelling language [Fisher 87] that allows models with subpart hierarchies, represents view dependent as well as object-centred information and also has a object property representation used for the model invocation process. It divides the matching process into two phases: First, an invocation network is called to select and weight a set of hypotheses based on evidence from extracted image features, or based on association with past matches; verification of the invoked hypotheses follows in the second phase of the matching by computing their pose in 3-D and providing a possible explanation for the match by predicting which features represented were visible, occluded or missing. Missing evidence is searched for (IMAGINE I).

The 3-D POLY system [Chen & Kak 89] was designed for the purpose of recognizing objects in the presence of occlusion and in scenes with cluttered backgrounds. The main novel feature of this system is a new data structure, called the feature sphere, that provides a way to efficiently assign and recover features from it improving the time complexity of the search process. They have designed a “principal direction” constraint whereby from knowing a first transformation hypothesis Tr, principal directions $\hat{\phi}$ are calculated as: 1) For primitive surfaces or curves, S, $\hat{\phi} = \mathbf{R}^{-1} * v(S)$, ($\mathbf{R}$ rotation component of Tr, $v(S)$ orientation direction of feature S); 2) For features, S, such as points (or spherical surfaces), $\hat{\phi} = Tr^{-1} * p(S)$, ($p(S)$ position vector of feature S). Matching was organized by first generating hypotheses using local feature sets (similar to [Oshima & Shirai 83] and [Bolles & Horaud 86]), and subsequently pairing them in an Interpretation Tree search with geometric and its principal direction constraints.

Another surface-based object recognition system was proposed by [Fan 90]. The system automatically produced a symbolic description of the objects in the scene in terms of their visible surface patches. After detecting jump, limbs, and crease boundaries on the range images the surfaces were segmented by joining these boundaries. The segmented representation is a graph with nodes having surface patches information and links representing relations such as occlusion and connectivity. A model is represented by a set of such descriptions from multiple (2-6) viewing angles. Matching is divided into three modules: "Screener", "Graph Matcher", and "Analyzer". In the "Screener" module a list of candidates is found by computing coarse differences on the nodes. The "Graph Matcher" module uses partial constraints and compares the candidates through a graph search. Finally the "Analyzer" module tries to reevaluate possible errors with previous matching by merging unmatched objects with matched ones to refine the match, or rearranging graph descriptions acquired for further search.

[Flynn & Jain 91] described the BONSAI system which recognizes 3-D objects from range images using a CAD package to design the models with surfaces. BONSAI represented 3-D objects using planes, cylinders, and spheres as primitives. A region-based segmentation strategy that applies a cluster analysis was used to separate large patches in the data. Classification of the patches into three types (planar, cylindrical, and spherical) follows with its output going to the matching module. The matching strategy uses unary and binary geometric constraints derived from the CAD models as pruning constraints in an interpretation tree search. [Flynn & Jain 91] reports results using a variety of range images of 3-D objects with good performance on accuracy both in the identification and localization stages of recognition.

[Kim & Kak 91] proposed a different strategy than the hypothesize-and-verify to recognize objects from large model libraries. Objects are represented by attributed surface features (planar, cylindrical, spherical, conical). Matching is organized as a bipartite matching problem, which means that given two sets of features represented by graphs (one from the scene, one from the models) assign for each node in one set a node from the other set. To enforce relational constraints and prune the search for the matching, discrete relaxation is used as the next filter stage. However, achieving a complete matching in this framework is not sufficient for solving object recognition because from the many potential complete matchings, only those that can be verified by computing feasible transformations are actually acceptable solutions. The pose estimation procedure used in the verification is similar to the technique of [Faugeras & Hebert 86].

[Hughes 91] addressed the problem of recognizing instances of 3-D curved objects with variability in scale but represented as a class with a parameterized model. The modelling primitives used are qualitative surface patches defined by the gaussian and mean curvatures of the patch (8 types), which are also extractable from the range data and given to the matching as pre-segmented data. Matching is organized as an Interpretation Tree search with geometric and scale constraints in two stages: In the first stage, after a corresponding pair is hypothesized, an initial estimate of the stretch deformation is computed and models are then selected on that basis for further matching; in the second phase a search through the chosen models is performed enforcing geometric constraints to identify the class of the object. The rigid transformation is computed at the end to strictly locate the object, estimate the model parameters, and compare its matched structure to the data.

[Stein & Medioni 92] proposed a geometric hashing algorithm for recognition of 3-D curved objects that introduces two different primitives for matching: A small surface patch called “splash”, which is a local gaussian map of the distribution of surface orientation along a geodesic circle; and a 3-D curve which is a polygonal approximation of the depth and orientation discontinuities of the object. The idea is to capture surface structure information through the polygonal approximations of “splashes” by computing and encoding 3-D super segments using: 1) curvature and torsion angles of a 3-D super segment; 2) maximum distance from the start of mapping; 3) radius of the “splash” surface; 4) cardinality of the 3-D super segments. The objects are represented also with these primitives and are arranged in a hash table for accessibility from a large database of objects. Recognition is then approached by first extracting the "splashes" and 3-D super segments from the range image, and then searching for correspondent keys into the hash table of objects' models. For verifying the hypotheses, a rigid transformation is computed for each hypothesis removing those whose transformation is not found. The remaining hypotheses are tested with geometric constraints and clustered into different sets. Pose estimation of the final hypotheses is calculated by least-squares to locate the solution.

2.3 Approaches using Volumetric Modelling Primitives

This section reviews approaches for 3-D object recognition using volumes as modelling primitives. Volumetric primitives are more complex than other representations, relating a single structure to a conceptually richer region in space. Models consisting of volumetric primitives are therefore low in complexity, and very flexible to compose, making the search process faster and relaxing the reliance on verification. The approach we propose in this thesis uses volumetric primitives, particularly superellipsoids, to model and describe objects in the scene. The works reviewed here in this section are conceptually closer to ours than those discussed in Sections 2.1 and 2.2. Table 2.3 shows their main properties succinctly.

[Nevatia & Binford 77] used generalized cylinders as modelling primitives for performing object recognition. The description is constructed as a relational graph of each view, using generalized cylinders to represent elongated parts of the object. Ribbons are used to help infer the generalized cylinders from the image. An indexing scheme is used by associating to each distinct piece of the object a three-bit code to help speed the search process in the matching. They use a relational graph matching approach organized as follows: A set of hypothesized models is chosen by comparing ribbons extracted from the scene to ribbons represented in the models, then, for each of the hypothesized models, all the extracted features are compared against the properties from the model such as type and connectivity relations. An accepted match would be one with as many model features present in the scene as possible, however with the excluding condition that for every feature in the scene graph there should be one correspondent found in the model.

Author	Sensing Modality	Model Representation	Matching Strategy
[Oshima & Shirai 83]	range	planes, spheres and cylinders	selection of a "kernel" (data-driven) followed by a model-driven search for adjacent regions of the "kernel"
[Bhanu 84]	range	planes	stochastic relaxation
[Grimson & Lozano-Perez 84]	range	planes	search (geometric constraints)
[Faugeras & Hebert 86]	range	quadrics	search (rigidity constraints)
[Bolles & Horaud 86]	range	planes and cylinders	search (local-feature-focus)
[Jain & Hoffman 88]	range	3-D surface patches (3 qualitative types)	rule-based (evidence attached to scene and model features)
[Fisher 89]	range	3-D surface patches (7 qualitative types)	constraint propagation and search
[Chen & Kak 89]	range	quadrics and spheres	search (hashing on orientation of features)
[Fan 90]	range	quadrics	attributed graph matching
[Flynn & Jain 91]	range	planes, cylinders and spheres	search (geometric constraints)
[Kim & Kak 91]	range	lines and quadrics	graph bipartite matching followed by discrete relaxation to prune the search space
[Hughes 91]	range	3-D surface patches (8 qualitative types)	search (geometric constraints)
[Stein & Medioni 92]	range	"splash" (3-D supersegment) and 3-D curves	hashing on the 3-D supersegments (curvature and torsion)

Table 2.2: 3-D Object Recognition Approaches using Surface Primitives

[Marr & Nishihara 78] proposed a hierarchical volumetric representation for modelling shapes like the human body and animals, which relies on finding natural axes of the shapes. They outline important criteria on representing three-dimensional objects. A fully working recognition system was not reported by the authors, but they describe a process of how recognition could be approached using their hierarchical description of objects. After the recovery of the cylinders from the image a linear search through indexed models (indexed by categories of shapes based on connectivity of the cylinders) is performed to achieve recognition. Constraints on the position of the axes are then checked using a relaxation algorithm. The approach as presented is incomplete, and limited in scope specially because it does not address the problem of how to extract the hierarchical description from the image for other sample objects, and the matching as proposed by them is sufficient only for a rough labeling of the object in the category with no further verification. However, it raises several questions that are needed to be considered for 3-D object recognition. Their main contributions were to outline a theory of model-based object recognition, and to give specific criteria on how to judge a representation for Computer Vision purposes.

The ACRONYM system [Brooks 81] was developed to identify and locate classes of 3-D objects from 2-D intensity images. The objects ACRONYM is able to recognize are aircraft observed from an aerial image of an airport. Image description is based on ribbons (which are a 2-D representation for the observed shape of generalized cylinders in 3-D), and ellipses. The objects are represented by an object graph, where for each object the nodes include generalized cylinder descriptions and subpart relations are indicated by arcs. A restriction graph is also built indicating a model class hierarchy based on constraints on the objects' quantifiers. Matching in the ACRONYM system is performed at two levels. First, a prediction graph is built using a rule-based module to identify visible contours of model surfaces and these predictions are compared to the ones present in the observation graph. At the second level, the ribbons matched are checked for consistency of the 3-D constraints specified both in the object and the restriction graphs.

[Mulgaonkar *et al.* 84] presented a technique to match "rough" relational models composed of "sticks, plates, and blobs". They do not address segmentation, and consider acquiring the data pre-segmented and with perspective projection. They recovered the camera parameters considering three possible explanations for the shapes: a sphere (blob) projects as an ellipse, a stick as a line or vanishing point, and a circle as a plate. Instead of using symbolic constraints as [Brooks 81] did they use numerical calculations to predict the appearance of subparts of the objects (rough objects/models). The parts of the objects are not allowed to rotate. The matching is a relaxation process with shape, view, and relational constraints. Input is given as 2-D views, either generated from models or taken from digitized photographs of furniture objects.

[Pentland 87] proposed the use of superquadrics as basic part structures of 3-D objects that could be recovered from range images by a process of optimization. Pentland described his approach as one of “general purpose” recognition, in which after having chosen a shape vocabulary (set of superquadrics) recognition is achieved by fitting a superquadric to the data acquired. This fitting is a gradient descent optimization procedure evaluated by a goodness-of-fit functional which determines the “best” model for the data. The model is one part only model, or any superquadric, there are no database of objects to compare the recovered object with. In a sense it is not a truly recognition approach since it does not address the problem of corresponding objects recovered from the scene with objects in a database represented with similar features. It showed though a way to recover superquadrics from the data, which are suitable modelling primitives both for computer vision and computer graphics. Others, ([Solina & Bajcsy 90][Gross & Boult 88]), have used superquadrics as modelling primitives in computer vision however addressing the problem of recovery and not matching. The approach presented in this thesis is to our knowledge the first attempt to build a model-based recognition system using superquadrics in the recovery and in the matching processes.

The PARVO system [Bergevin & Levine 89] was an attempt to recover a set of volumetric primitives from labelled line drawings and identify the objects using coarse qualitative models. They referred to “generic object recognition” as model-based object recognition not by means of accurate geometric models but by coarse models representing classes of objects. In their system Bergevin and Levine rely strongly on the pre-labelled line drawing of the object given as input, and they proceed to construct a graph structure of the object in the scene by labelling its parts using a set of 11 geons and including adjacency and aspect ratio information. Results are shown for perfect line drawings of a variety of objects. Recognition is considered achieved after graph matching is performed on the database without verification of pose or other quantitative properties of the objects.

[Dickinson *et al.* 92b] described a technique to perform recovery and recognition of 3-D objects from a single 2-D image. For a given recognition domain consisting of a database of objects, it first selects a set of object-centered 3-D volumetric modelling primitives (a subset of geons) to construct the objects. A set of aspect-graphs of the primitives is generated using a CAD system. The aspects are used to model the parts of objects. Testing of the hypotheses consists of a topological verification of the recovered graph, rather than a geometric verification of image features. Pose estimation is not addressed and recognition is accepted after the graph correspondences are identified.

Author	**Sensing Modality**	**Model Representation**	**Matching Strategy**
[Nevatia & Binford 77]	range	generalized cylinders	relational graph matching
[Marr & Nishihara 78]	2.5D	generalized cylinders	linear search through all models followed by relaxation to constrain position of the axes
[Brooks 81]	intensity	generalized cylinders	constraint propagation
[Mulgaonkar *et al.* 84]	2-D prepared views, intensity	"sticks, plates and blobs" (relational models)	constraint propagation
[Pentland 87]	range	set of superquadrics	explain the data by fitting "the best" superquadric on a region (*i.e.* minimizes a functional). No pre-defined database of models.
[Bergevin & Levine 92]	intensity (partitioned line drawing given)	symbolic geons	linear search into indexed database
[Dickinson *et al.* 92b]	intensity	hierarchical aspect-graph rep. of volumetric primitives (geons)	hashing on the number of recovered volumetric primitives followed by a weighted graph matching

Table 2.3: 3-D Object Recognition Approaches using Volumetric Primitives

Chapter 3

Segmentation into Volumetric Parts using Dynamic Link of Discontinuities

One of the key questions in Computer Vision research is how to segment a scene into meaningful chunks which have stable visual interpretation and can be extracted in a reliable way. This segmentation process is especially important as a first stage of a system which aims to reason about and recognize 3-D objects based on the notion of prototypical parts. This chapter reports on the segmentation approach undertaken in this thesis.

To achieve the desired segmentation we present a multi-stage approach to identifying separate parts of natural articulated objects by relying on stable estimates of the differential structure of the object's surfaces, followed by a dynamic grouping of the detected discontinuities on the surface and a fitting of deformable superquadric models to the segmented regions.

The final aim of the segmentation is to get a set of 3-D deformable superquadric models, where each model of the set is related to only one part of the object. Our assumption is that three types of discontinuities (depth, concave orientation, and loci of negative minima of the Principal Directions) detected upon the local differential structure of the object's surface can be grouped together so as to become the boundaries of salient parts of an articulated 3-D object. This grouping is formulated as an energy-minimization process of deformable curves, wherein initially unit-length curves are placed at the discontinuity points and then are attracted to paths of minimum change in the Principal Directions (around a fixed neighbourhood) until they reach other discontinuities. After this stage, the set of final curves will indicate the regions on the object's surface to be represented as separate parts. We propose that globally deformable superquadrics can capture this notion of an object's parts well. Therefore, in our solution we fit deformable superquadrics to each of the segmented regions to get a final extracted model of the object suited for identification of its overall shape and position. Previous results on the segmentation method presented in this chapter were reported in [Borges & Fisher 93].

The approach uses a mix of techniques which provide a tailored solution to parts-based segmentation of 3-D objects. The overall structure of the segmentation approach bears similarities to work done by [Ferrie *et al.* 89], but it differs from their work by the inclusion of discontinuity preserving conditions for smoothing and curvature consistency calculations, dynamic grouping of all the detected discontinuities and by the use of deformable superquadric models for representing the objects' parts.

In total there were eleven (11) objects tested, with seventeen (17) test scenes. Chapters 3, 4, 5, and 6 show results on four different objects and scenes, the remaining of the test scenes with their respective results are included in Appendix F. All the objects selected for testing were chosen from wooden, and plastic toys with animal-like forms because they represent a reasonable set of 3-D models with many curved surfaces, symmetry relative to the main view, a fair amount of similar parts between the objects in the set making them more difficult to be differentiated, and some articulated parts. The set contains 3-D objects with a good deal of form variation for testing, and for two objects ("horse" and "pig") four different range images were taken of each by changing their viewing positions.

Figure 3.1 shows a functional block diagram of the segmentation approach.

The following sections are divided according to each stage of the segmentation. Results are shown in Section 3.6, and Section 3.7 gives a summary of the chapter.

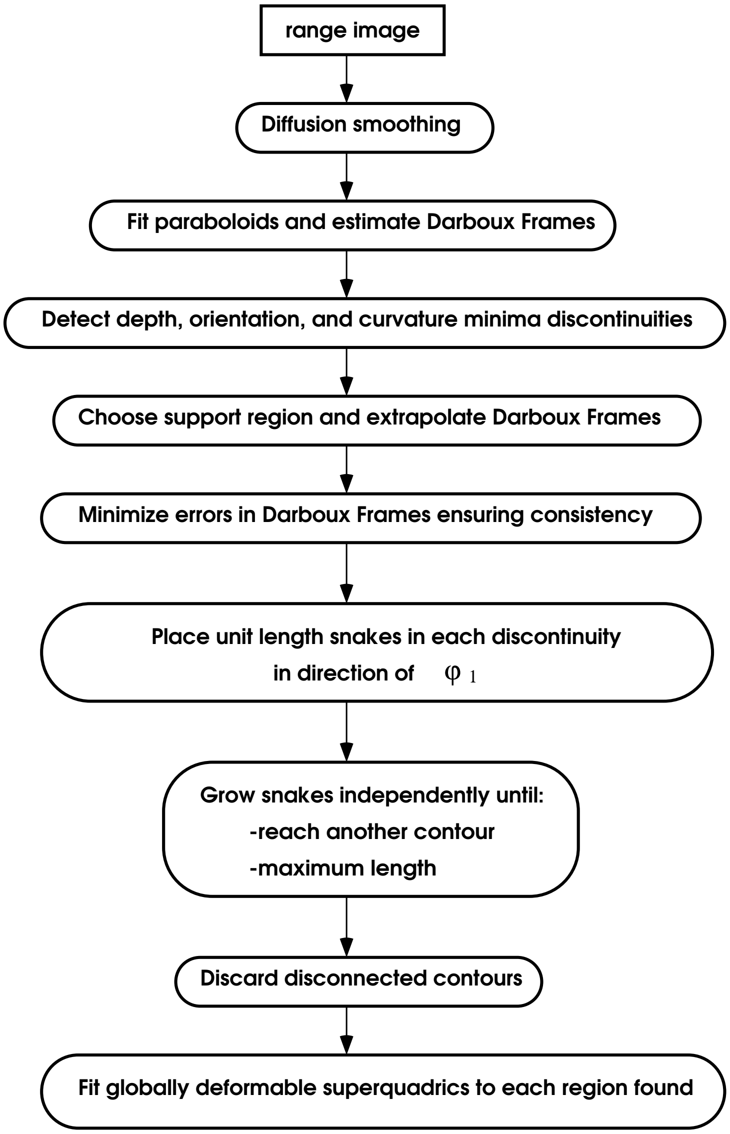


Figure 3.1: Functional block diagram of the segmentation approach used to extract volumetric parts of the object

3.1 The Problem of Segmentation into Parts: Which Discontinuities?

Defining what constitutes a part of an object is a very important and difficult problem. We perceive regularities in our visual world, and these regularities help us to organize chunks of an image into meaningful parts. A set of these parts would constitute a rough model of the object.

Important evidence for defining the extent of an object part is some sort of discontinuity on the surface. Two types of discontinuities useful here are depth discontinuities and concave orientation discontinuities. As first argued by Hoffman and Richards [Hoffman & Richards 85], another important principle for parts decomposition is the *transversality regularity*, which refers to the loci of negative minima of each principal curvature as belonging to part boundaries.

In the segmentation approach presented here we detect three types of discontinuities and place upon them deformable contours which will subsequently grow following a tangent field, until a closed boundary is achieved. The next section describes how the discontinuities are detected, and subsequent sections deal with how to link them in order to achieve closed contours, and how to derive volumetric models from the regions found.

3.2 Detecting Discontinuities

Depth discontinuities are detected in the range image by first estimating the value of the directional derivative

$$d(\mathbf{p}, \mathbf{n}) = \frac{\partial u}{\partial \mathbf{n}}(\mathbf{p}) \tag{3.1}$$

at each point $\mathbf{p}$ of the image for eight pre-defined directions of $\mathbf{n}$, and then comparing these values with a given threshold τ_d. For the range images used in the experiments throughout this thesis, τ_d was between $(5 < \tau_d < 15)$.

Orientation discontinuities are defined as the discontinuity points for the tangent plane, and their detection is also relative to a pre-defined threshold, τ_o. Given the following parameterization of the surface, $S(x, y) = (x, y, h(x, y))$, and considering a point $\mathbf{p}$ on

S, the orientation discontinuities on S are the points $\mathbf{t}$ in a neighbourhood of $\mathbf{p}$ (in our case 5x5) that satisfy

$$| \, \mathbf{n}(\mathbf{p}) \cdot \mathbf{n}(\mathbf{t}) \, | \quad < \quad \tau_o \tag{3.2}$$

where $\mathbf{n}(\mathbf{p})$ is the normal to S on $\mathbf{p}$. In the experiments throughout this thesis τ_o used was between $(0.7 < \tau_o < 0.9)$.

The *negative minima of curvature* discontinuities are defined as: The points where the directional derivatives of the principal curvatures ($\kappa_{1,2}$) along each principal direction ($\psi_{1,2}$) respectively satisfy the following [Hoffman & Richards 85],

$$\frac{\partial \kappa_1}{\partial \psi_1}(x, y) = 0 \quad \wedge \quad \kappa_1(x, y) < 0 \tag{3.3}$$

$$\vee$$

$$\frac{\partial \kappa_2}{\partial \psi_2}(x, y) = 0 \quad \wedge \quad \kappa_2(x, y) < 0 \tag{3.4}$$

The negative minima of curvature discontinuities play an important role in the segmentation of surfaces according to the principle of transversality regularity they suggest where two concave parts have joined. [Hoffman & Richards 85].

3.3 Calculating the Differential Structure of a 3-D Object's Surface

In this section we present the approach we used for calculating the differential structure of the object surface, with the aim of achieving accurate and stable estimates of the Darboux Frames.

There are three basic stages in our approach.

1. Range images are acquired using a laser-stripe based range sensor developed in our laboratory [Trucco & Fisher 94], and then submitted to pre-processing in the *rangeseg* system [Trucco & Fisher 92] in order to reduce noise by diffusion smoothing, which uses the depth and orientation discontinuities found in the previous section to avoid introducing false curvature regions.

2. Local fitting of paraboloid patches is performed using a transformation from global to local coordinates, which ensures good first estimates of the Gauss and

Darboux Frames. The 3-D edge map from Section 3.2 is used to avoid fitting the quadric patches over the discontinuities.

3. The last stage involves a minimization of a functional over the Darboux Frames estimated in the previous stage. This process starts by extrapolating the values in local coordinates to the values of the Darboux Frames at a point in the centre of a fixed neighbourhood. It then constructs a functional to ensure small errors (in the least square sense) and consistency of the measures.

The following subsections deal with the second and the third stage in detail.

3.3.1 Fitting Paraboloid Patches in Local Coordinates for First Estimation of Darboux Frames

The method consists of building a local coordinate frame $(\mathbf{u}, \mathbf{v}, \mathbf{n})$ at an inspection point on the surface of the object, then it takes all surface points (x, y, z) in its neighbourhood and transforms them into $(u, v, h(u, v))$. Next the values for the local curvatures and Darboux Frames are calculated from an approximation to $h(u, v)$.

The steps of the method are the following (see Appendix A for the full set of equations):

- First, a normal vector $\mathbf{n}$ is calculated at the surface point, which characterizes the tangent plane at that point.
- Two vectors $\mathbf{u}$ and $\mathbf{v}$ are chosen on the tangent plane to become the u and v axes of the parameterization.
- A transformation from global (x, y, z) to local coordinates (u, v, n) is calculated.
- The surface patch will then be approximated (using a paraboloid surface patch) as

$$S(u,v) = \quad (u, v, h(u,v)) = \quad (u, v, au^2 + 2buv + cv^2) \tag{3.5}$$

or choosing a neighbourhood of n points and writing it in a matrix form

$$\mathbf{Ax} = \mathbf{b}$$

$$\begin{bmatrix} u_1^2 & 2u_1v_1 & v_1^2 \\ u_2^2 & 2u_2v_2 & v_2^2 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ u_n^2 & 2u_nv_n & v_n^2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \\ \cdot \\ \cdot \\ \cdot \\ h_n \end{bmatrix} \tag{3.6}$$

- which has the generalized inverse solution:

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \tag{3.7}$$

Having the parameterization in $S(u,v)$ the next step is to calculate the Gaussian curvature K, the Mean curvature H, Principal curvatures $\kappa_{1,2}$, and Principal directions $\psi_{1,2}$ (see Appendix A for equations).

3.3.2 Enforcing Curvature Consistency for Achieving Better Accuracy of Darboux Frames

For accurate and reliable estimates for the set of surface curvatures, the first estimation procedure based on a local coordinate fit of the paraboloid patches is not good enough. The method we use here was first proposed by [Sander & Zucker 90], and it works as a functional to minimize errors and improve consistency on the curvature values. Our objective is to acquire a reliable and accurate set of Darboux Frames of the object surface for the segmentation, and this procedure achieves just that.

The method works as follows:

1. First, for all points **p** (for which there is already an estimated curvature set) in a fixed neighbourhood (*e.g.* 5x5), determine the support region $\pi^{(i)}$ as:
   For each point $\mathbf{q}_\alpha$ within a fixed distance from **p**, choose the ones whose local information at $\mathbf{q}_\alpha$ represents a quadric surface S_α which contains **p**. The $\mathbf{q}_\alpha$ points which satisfy this condition are in the support neighbourhood.

2. For all points $\mathbf{q}_\alpha$ in the support neighbourhood $\pi^{(i)}$ compute the frames $\Gamma^i_{p_\alpha}$ at point **p** from the frames Γ^i_α at $\mathbf{q}_\alpha$ (see Appendix A for equations).

3. Given the set of frames $\Gamma^i_{p_\alpha}$, compute the best estimate of Γ^i_p that minimizes variation according to the following constraints:

$$(\mathbf{n}_p \cdot \mathbf{n}_p) = 1$$

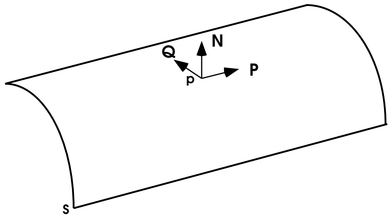


Figure 3.2: Local Darboux Frame calculation. **N** is the normal to S in p, and **P** and **Q** are derived to create a local coordinate system in p from **N**.

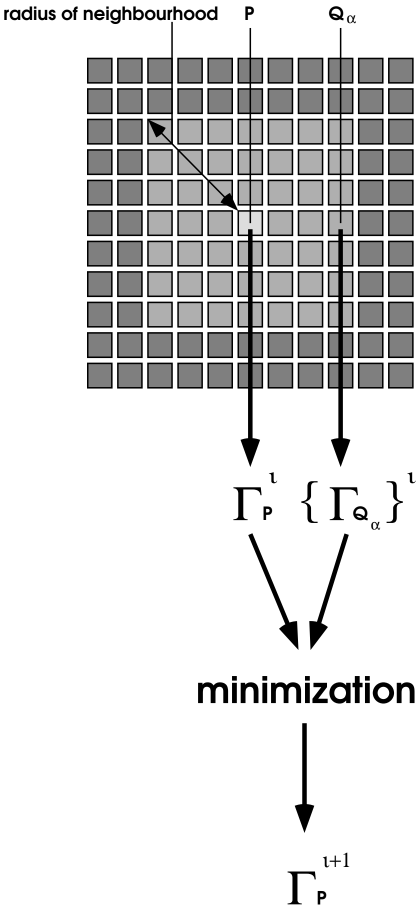


Figure 3.3: Schematic view of the minimization of Darboux Frames.

$$\begin{aligned} (\psi_{1p} \cdot \psi_{1p}) &= 1 \\ (\psi_{1p} \cdot \mathbf{n}_p) &= 0 \end{aligned} \tag{3.8}$$

This yields two terms to be minimized:

$$\begin{aligned} E_1 &= \sum_{\alpha=1}^{n} \|\mathbf{n}_p - \mathbf{n}_{p_\alpha}\|^2 + (\kappa_1 - \kappa_{1p_\alpha})^2 + (\kappa_2 - \kappa_{2p_\alpha})^2 + \\ & \quad \lambda((\mathbf{n}_p \cdot \mathbf{n}_p) - 1) \qquad (3.9) \\ E_2 &= \sum_{\alpha=1}^{n} \left[1 - (\psi_{1p} \cdot \psi_{1p_\alpha})^2\right] \qquad (3.10) \end{aligned}$$

The global measure is:

$$R_s^i = \sum_j E_{1j}^{\,i} + E_{2j}^{\,i} \qquad \mathbf{p}_j \in s. \tag{3.11}$$

4. Use minimization to compute new values of Γ_p^{i+1} at $\mathbf{p}$.

5. Compare R_s^i with R_s^{i-1}. If the difference is greater than a threshold (in our case 10^{-3}), return to step 1.

Figure 3.3 shows an schematic view of this minimization procedure. The next section will address the problem of how to link the discontinuities found here, and in Section 3.2, in order to provide perceptually significant segmentation of the object into parts.

3.4 Dynamic Linking of Discontinuities

All of the three kinds of discontinuities (depth, concave orientation and the negative minima of curvature) are important in determining the extent of an object's part. The problem we face now is how to combine them to isolate the most salient parts the objects suggest.

A suitable framework for deriving these contours is the notion of energy-minimizing curve fitting, or snakes [Kass *et al.* 87]. Here, fixed length snakes are placed initially at the discontinuity points detected above and they grow to follow the direction of a local tangent field [Zucker *et al.* 88] model derived from the differential structure of the 3-D surface.

The method can be divided in two stages:

First stage. Initially, for each discontinuity point (either depth, concave orientation, or negative of minima curvature) a potential field is created based on the position and the principal direction ψ_1 at that point.

Formally, as in [David & Zucker 90] the set of 2-D Gaussians $\{G_i : i = 1, ..., n\}$ is the set of generators of the potential field. At each point $i = (x_i, y_i)$, with principal directions ψ_{1i}, G_i is

$$\begin{aligned} G_i(x,y) &= -(W_E exp^{-\frac{(f_i(x,y)-x_i)^2}{\sigma_E^2}}).(W_B exp^{-\frac{(g_i(x,y)-y_i)^2}{\sigma_B^2}}) \\ &= -W_E W_B.exp^{(-\frac{(f_i(x,y)-x_i)^2}{\sigma_E^2}-\frac{(g_i(x,y)-y_i)^2}{\sigma_B^2})} \qquad (3.12) \end{aligned}$$

where, $f_i(x,y)$ and $g_i(x,y)$ are given by,

$$f_i(x,y) = x_i + (x - x_i)cos\psi_{1i} + (y - y_i)sin\psi_{1i} \qquad (3.13)$$

$$g_i(x,y) = y_i - (x - x_i)sin\psi_{1i} + (y - y_i)cos\psi_{1i} \qquad (3.14)$$

The parameter σ_E is called the extension parameter, and σ_B the blurring parameter. The weights W_E and W_B are positive numbers associated with the 1-D gaussians. For the experiments in this thesis, the values used were $\sigma_E = 2.75$, $\sigma_B = 0.6875$, $W_E = W_B = 1$.

The potential field U_{PT} is then given by the following sum of G_i

$$U_{PT}(x,y) = \sum_{i=1}^{n} G_i(x,y) \qquad (3.15)$$

One important feature of this method is that the valleys produced in the potential field distribution, Equation (3.15), represent the most probable (in the Wiener measure sense) sets of curves in agreement with the directions given [David & Zucker 90].

Second stage. In this part the method recovers the sets of curves embedded in those valleys.

Initially a unit-length deformable contour (a snake [Kass *et al.* 87]) is placed on each discontinuity point with the orientation of the principal direction ψ_1. The potential field is then calculated using Equation (3.15) for all the discontinuity points.

Independently from each other, all the initial snakes are allowed to grow following the potential field. The other parameters of the snakes are set to standard values as in the original algorithm of [Kass *et al.* 87]. Appendix B shows all the equations for the snake minimization procedure and the constants used.

Since our aim here is to find the closed contours of the regions, the following conditions control snake growth:

- Stop growth, if another snake at an adjacent pixel.
- If no snake is met, continue growing according to last direction until another snake is found.

3.5 Fitting Parametric Primitives to Segmented Regions: The Choice for Globally Deformable Superquadrics

Once a region can be successfully separated by the dynamic linking procedure of Section 3.4, the next step is to fit a volumetric model to this region. A volumetric primitive that would represent well the notion of part, both in natural and in man-made objects, is a superquadric [Pentland 86].

3.5.1 What is a Superquadric?

Superquadrics are a family of parametric shapes that can be globally deformed, and therefore represent a wide range of prototypical parts. They were introduced by Piet Hein [Gardner 65], and then first used in the fields of Computer Graphics and Computer Vision by [Barr 81] and [Pentland 86] respectively.

A superquadric surface can be defined by the following vector in 3-D space

$$x(\eta, \omega) \quad = \quad \begin{bmatrix} a_1 \cos^{\epsilon_1}(\eta) \cos^{\epsilon_2}(\omega) \\ a_2 \cos^{\epsilon_1}(\eta) \sin^{\epsilon_2}(\omega) \\ a_3 \sin^{\epsilon_1}(\eta) \end{bmatrix} \tag{3.16}$$

where,

$$-\pi/2 \leq \eta \leq \pi/2. \tag{3.17}$$

$$-\pi \leq \omega < \pi. \tag{3.18}$$

The parameters a_1, a_2, and a_3 define the size of the superquadrics in the coordinates x, y, and z respectively. ϵ_1 and ϵ_2 are the squareness parameters. The capabilities of the superquadrics for modelling can be enhanced by introducing global tapering and bending deformations to them. Also, since these deformations are global, the advantages of indexing of superquadrics in a parametric form are retained and a bigger vocabulary of shapes is introduced, whereas in using local deformations, as in [Terzopoulos & Metaxas 90], there is no longer the possibility of indexing. We formulate global bending and tapering deformations as in [Solina 87, Solina & Bajcsy 90], choosing the axis z as the main axis of the object and performing the deformations in x and y. (Appendix C shows tapering and bending deformation definitions).

3.5.2 Recovery of Superquadrics by Nonlinear Least Squares

For the purpose of recovering a superquadric surface from the data, Equation (3.16) can be manipulated in order to obtain the following implicit equation

$$\left(\left(\frac{x}{a_1}\right)^{2/\epsilon_2} + \left(\frac{y}{a_2}\right)^{2/\epsilon_2}\right)^{\epsilon_2/\epsilon_1} + \left(\frac{z}{a_3}\right)^{2/\epsilon_1} = 1. \qquad (3.19)$$

Based on this equation we can define the inside-outside function [Solina & Bajcsy 90]

$$\mathbf{F}(x,y,z) = \left(\left(\left(\frac{x}{a_1}\right)^{2/\epsilon_2} + \left(\frac{y}{a_2}\right)^{2/\epsilon_2}\right)^{\epsilon_2/\epsilon_1} + \left(\frac{z}{a_3}\right)^{2/\epsilon_1}\right)^{\epsilon_1}. \qquad (3.20)$$

Function (3.20) determines where a given point $[x, y, z]$ lies relative to the superquadric surface

- $\mathbf{F}(x,y,z) = 1$, point (x,y,z) is on the surface
- $\mathbf{F}(x,y,z) > 1$, point (x,y,z) is outside the surface
- $\mathbf{F}(x,y,z) < 1$, point (x,y,z) is inside the surface

Equation (3.20) defines the superquadric surface in an object centered coordinate system (x_s, y_s, z_s). By using a world coordinate system (with origin at (p_x, p_y, p_z)) and Euler angles (ϕ, θ, ψ) we can express the inside-outside function in general position

$$\mathbf{F}(x_w, y_w, z_w) = \mathbf{F}(x_w, y_w, z_w; a_1, a_2, a_3, \epsilon_1, \epsilon_2, \phi, \theta, \psi, p_x, p_y, p_z). \qquad (3.21)$$

The expanded function **F**, equation (3.21), has now 11 parameters excluding tapering and bending deformations. Once we include the deformations, the number of parameters to be recovered goes up to 15

$$\mathbf{F}(x_w, y_w, z_w) \quad = \quad \mathbf{F}(x_w, y_w, z_w; a_1, ..., a_{11}, b, \beta, T_x, T_y) \qquad (3.22)$$

where,

- (x_w, y_w, z_w) is a point on the surface of the superquadric
- $a_1, ..., a_{11}$ are the 11 parameters of an undeformed superquadric
- b, β are the bending parameters along axis z
- T_x, T_y are the tapering parameters along axis z

Following Solina [Solina & Bajcsy 90] we estimate the 15 parameters using the Levenberg-Marquadt method for nonlinear least squares minimization of the expression

$$\sum_{i=1}^{N} \left[\mathbf{R}(x_i, y_i, z_i; a_1, ..., a_{15})\right]^2 . \qquad (3.23)$$

where,

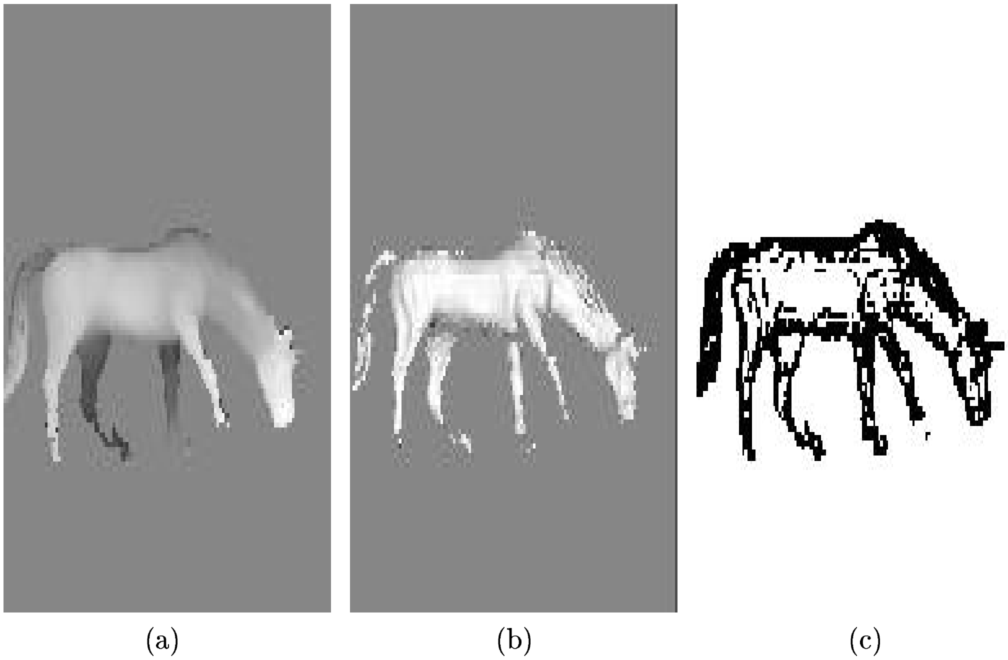
$$\mathbf{R} \quad = \quad \sqrt{a_1 a_2 a_3}(\mathbf{F} - 1). \qquad (3.24)$$

and N is the number of observed points. In our case, we use all of the 3-D data points in each region whose boundaries were previously processed by linking the discontinuities.

To include the deformations there has to be a priority in the order of recovery imposed on them. According to studies by Leyton [Leyton 87] the following order preserves better the structure of the object: shape, then tapering, bending, rotation and finally translation.

3.6 Experiments and Results

In this section we present results of the application of the segmentation approach described earlier on a set of range images of articulated objects. The images were acquired using the "Edinburgh University Laser Striper" [Trucco & Fisher 94], which is a triangulation range finder producing dense range images. The results are shown in different subsections according to each stage of the segmentation.



(a) (b) (c)

Figure 3.4: (a) Range image; (b) Pseudo intensity image; (c) Depth, orientation, and curvature minima discontinuities found in the range image.

3.6.1 Range images, Pseudo intensity, and Discontinuities Detected

Figures 3.4, 3.5, 3.6, and 3.7 show the range image acquired, a pseudo intensity image generated by cosine shading the range, and the corresponding discontinuities image found in the segmentation. The objects are plastic and wooden miniatures with complex curved 3-D surfaces and articulations. The objects are a "horse", a "cow", a "giraffe", and a "doll".

3.6.2 3-D Map and Principal Direction ψ_1 which Determines the Tangent Field

Figures 3.8, 3.9, 3.10, and 3.11 show sampled surfaces of the objects "horse", "cow", "giraffe", and "doll" respectively, with superimposed vectors indicating two stages (1) after first fitting, and 2) after consistency) of the computed principal direction ψ_1 for the surface.

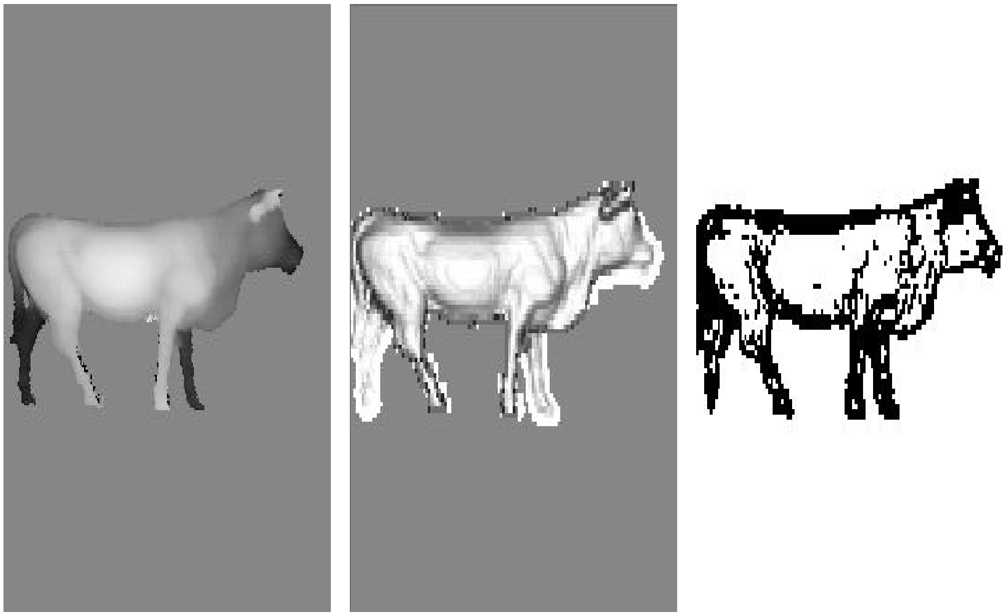

(a) (b) (c)

Figure 3.5: (a) Range image; (b) Pseudo intensity image; (c) Depth, orientation, and curvature minima discontinuities found in the range image.

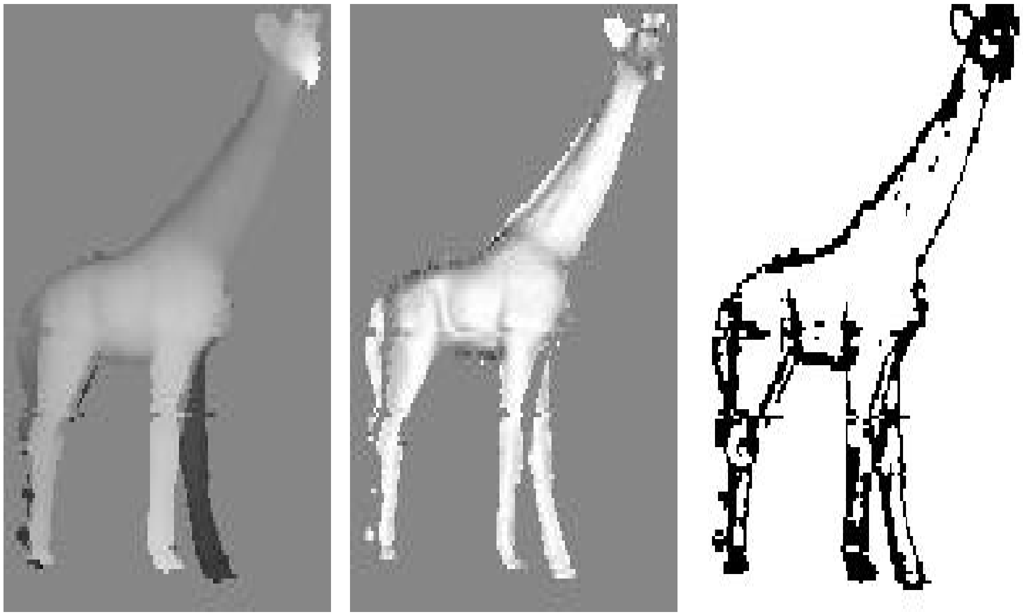

(a) (b) (c)

Figure 3.6: (a) Range image; (b) Pseudo intensity image; (c) Depth, orientation, and curvature minima discontinuities found in the range image.

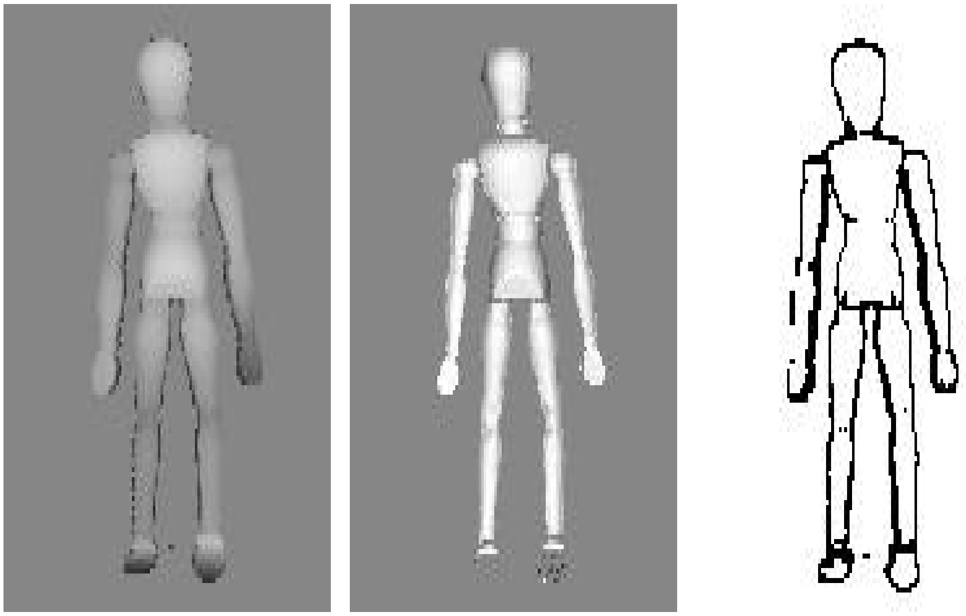


Figure 3.7: (a) Range image; (b) Pseudo intensity image; (c) Depth, orientation, and curvature minima discontinuities found in the range image.

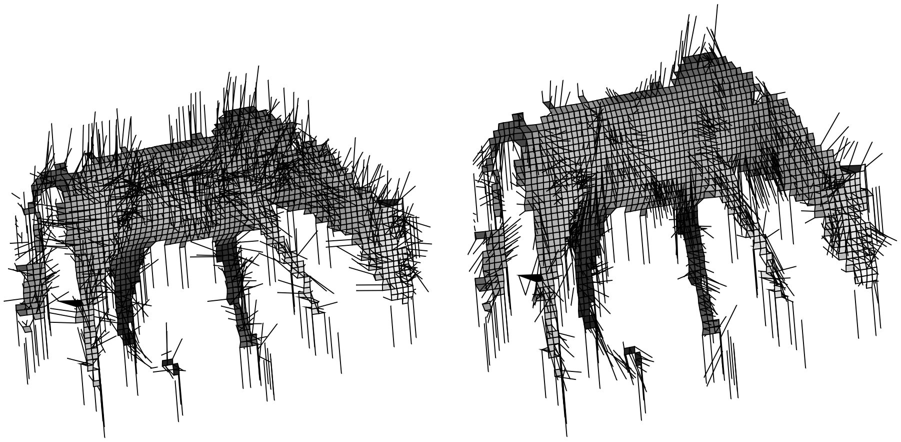


Figure 3.8: Sampled principal directions ψ_1 over a region of object "horse", (a) after first estimation using paraboloid fitting; (b) after running minimization to impose consistency.

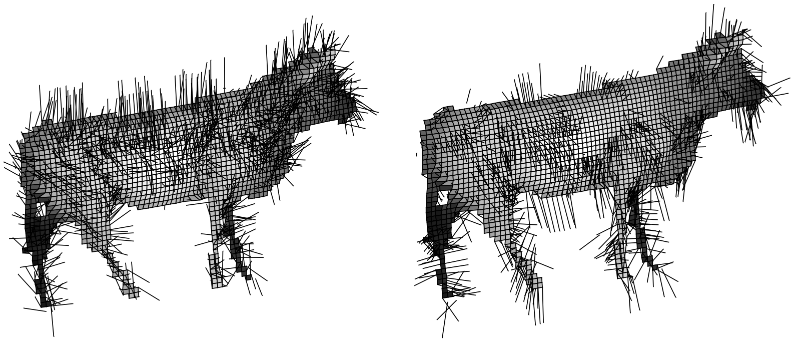

(a) (b)

Figure 3.9: Sampled principal directions ψ_1 over a region of object "cow", (a) after first estimation using paraboloid fitting; (b) after running minimization to impose consistency.

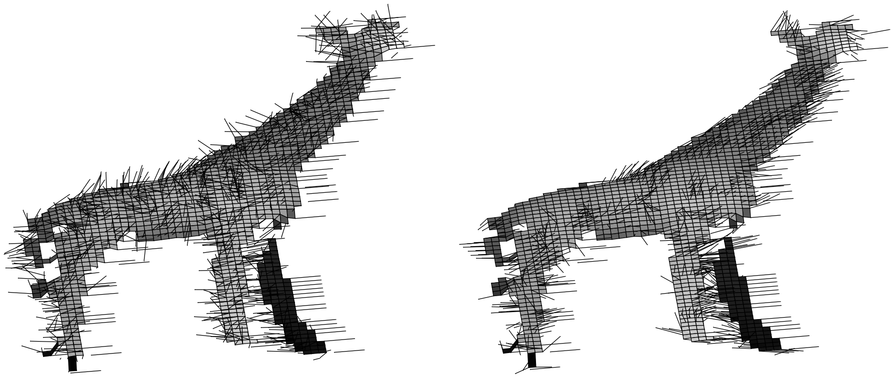
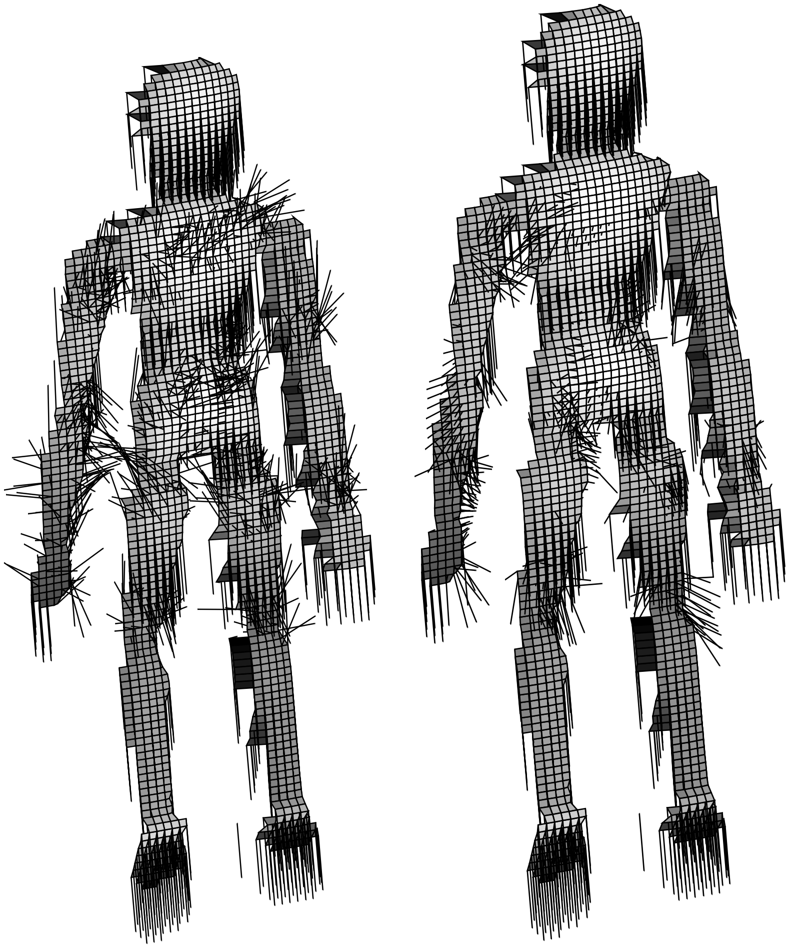

(a) (b)

Figure 3.10: Sampled principal directions ψ_1 over a region of object "giraffe", (a) after first estimation using paraboloid fitting; (b) after running minimization to impose consistency.



(a) (b)

Figure 3.11: (a) Sampled principal directions ψ_1 after first estimation (over region of object); (b) Sampled principal directions after running curvature consistency over first estimation (over region of object).

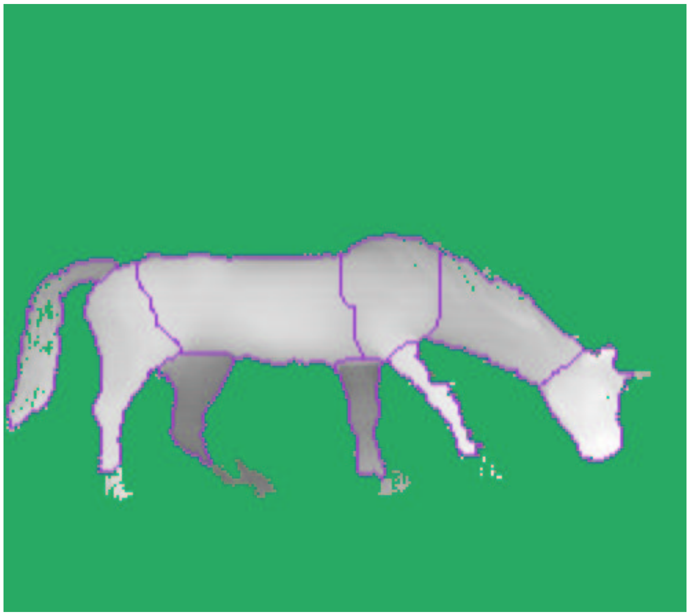


Figure 3.12: Regions found on object "horse" after run of detection and linking of discontinuities procedures. Contours which did not achieve closed boundaries were ignored.

3.6.3 Results on Linking the Discontinuities

Figures 3.12, 3.13, and 3.14 show final outputs of the discontinuities linking procedure with the successful regions found for the test objects.

3.6.4 Globally Deformable Superquadrics Fitted on the Segmented Regions

Figures 3.16, 3.17, and 3.18 show the superquadrics recovered for each separate region of the object "horse" overlayed on the data points, all superquadrics overlayed on the complete object, and the complete set of recovered superquadrics in two different positions, respectively.

Figures 3.19, 3.20, and 3.21 show the superquadrics recovered for each separate region of the object "cow" overlayed on the data points, all superquadrics overlayed on the complete object, and the complete set of recovered superquadrics in two different positions, respectively.

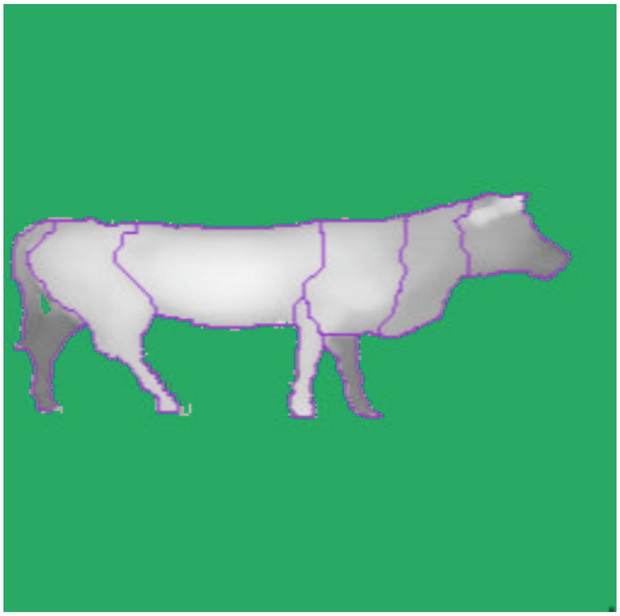


Figure 3.13: Regions found on object "cow" after run of detection and linking of discontinuities procedures. Contours which did not achieve closed boundaries were ignored.

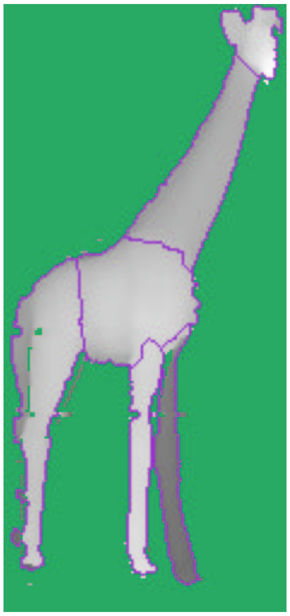


Figure 3.14: Regions found on object "giraffe" after run of detection and linking of discontinuities procedures. Contours which did not achieve closed boundaries were ignored.

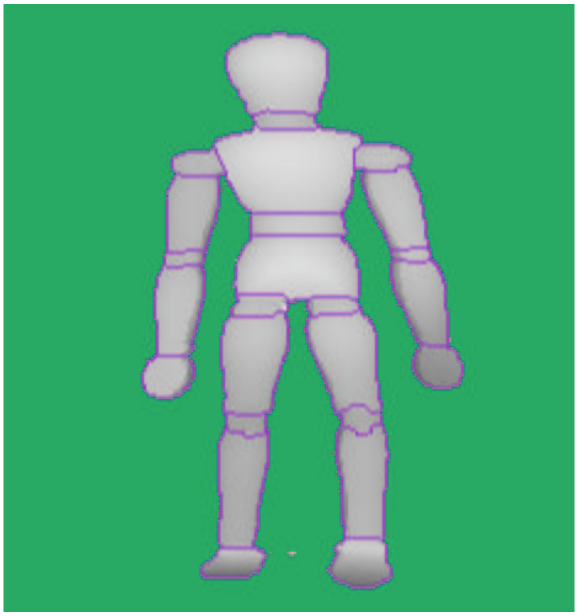


Figure 3.15: Regions found on object "doll" after run of detection and linking of discontinuities procedures. Contours which did not achieve closed boundaries were ignored.

Figures 3.22, 3.23, and 3.24 show the superquadrics recovered for each separate region of the object "giraffe" overlayed on the data points, all superquadrics overlayed on the complete object, and the complete set of recovered superquadrics in two different positions, respectively.

Figures 3.25, 3.26, and 3.27 show the superquadrics recovered for each region of the object "doll" overlayed on the data points in separate, all superquadrics overlayed on the complete object, and the complete set of recovered superquadrics in two different positions, respectively.

3.7 Discussion and Criticisms

The most important deficiency of the segmentation presented here is stability. The stages of discontinuity detection and region closing, and of superquadric parameter estimation rely to a large extent on the view upon which the range image is taken. It is assumed a view where most of the object's surface is seen can always be obtained, however some perturbations of small order on acquiring the data have to be taken into

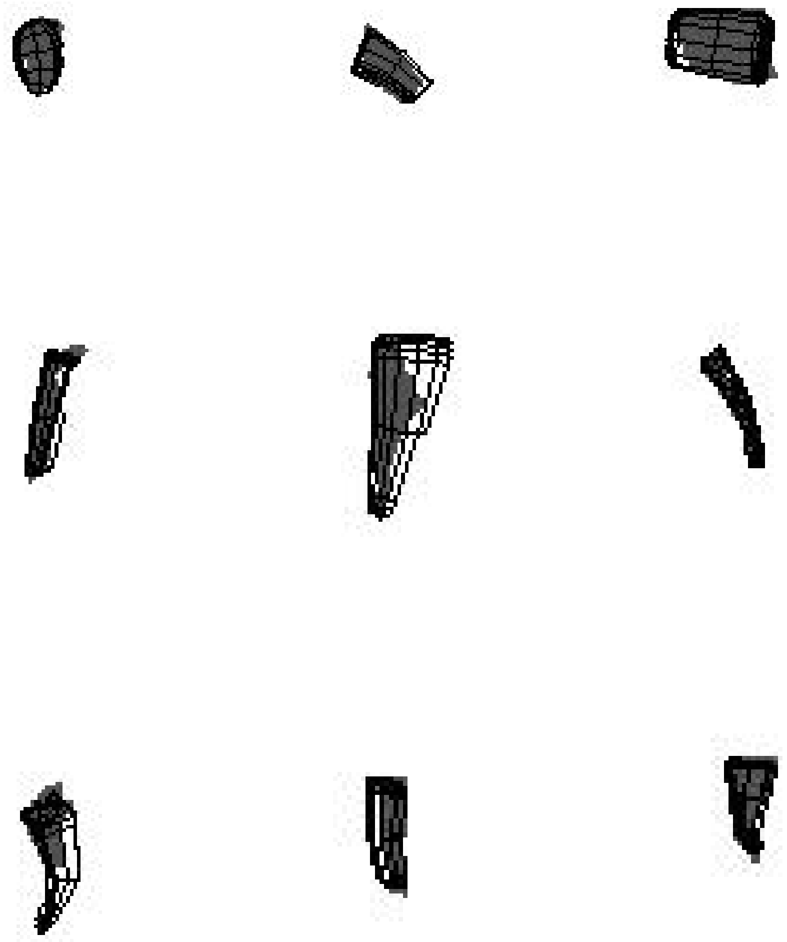


Figure 3.16: Superquadrics wire-frames shown on top of each region separately for object “horse”.

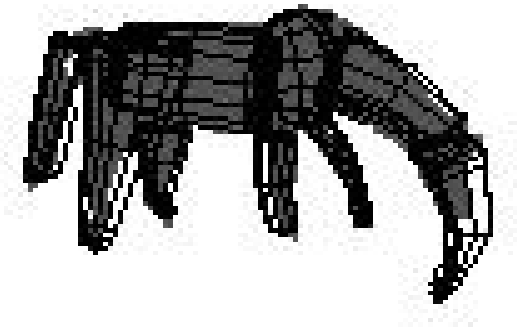


Figure 3.17: Superquadrics wire-frames shown on top of fitted regions for object "horse".

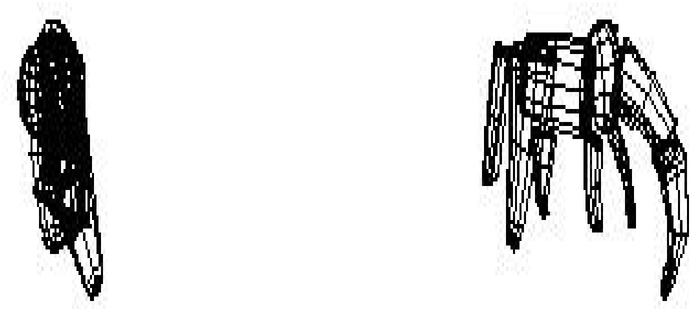


Figure 3.18: Superquadrics recovered from data object "horse" shown in two different positions.

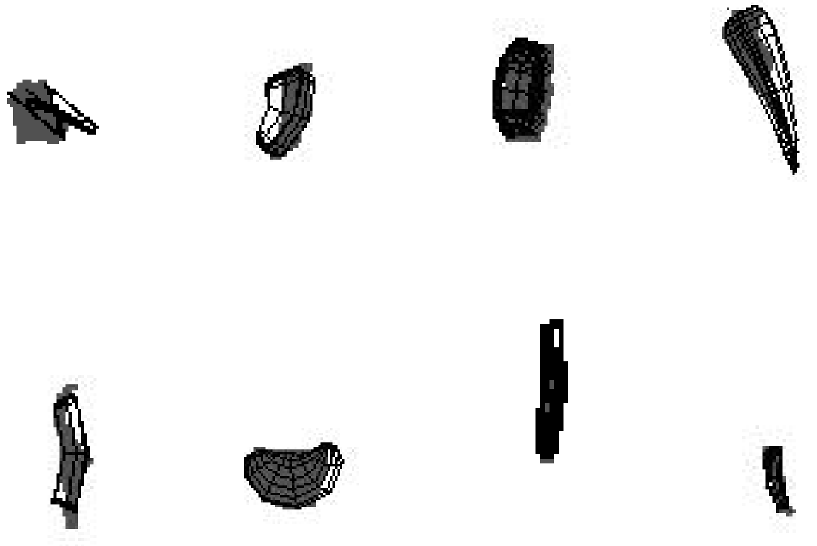


Figure 3.19: Superquadrics wire-frames shown on top of each region separately for object “cow”.

Figure 3.20: Superquadrics wire-frames shown on top of fitted regions for object “cow”.

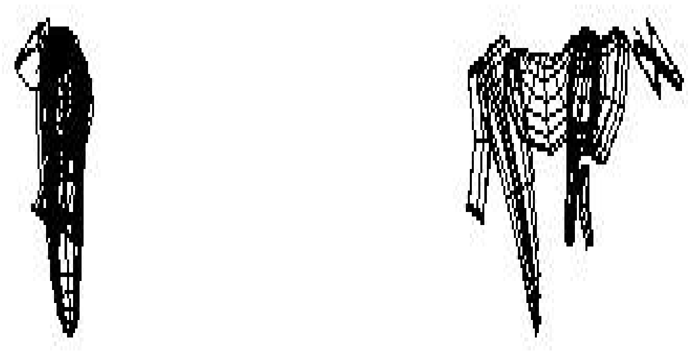


Figure 3.21: Superquadrics recovered from data object "cow" shown in two different positions.

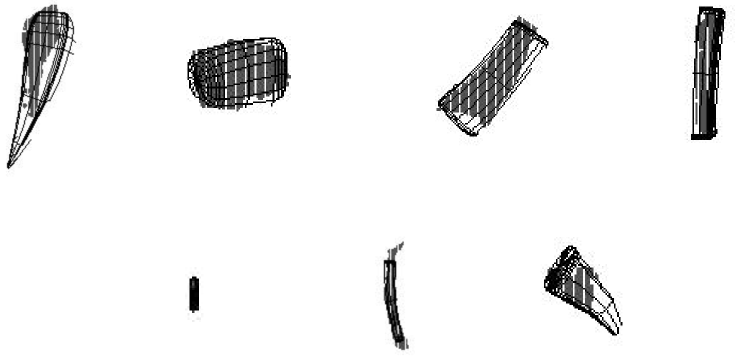


Figure 3.22: Superquadrics wire-frames shown on top of each region separately for object "giraffe".

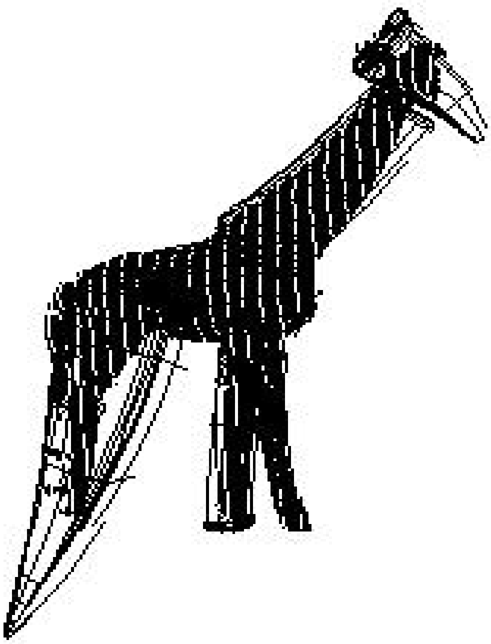


Figure 3.23: Superquadrics wire-frames shown on top of fitted regions for object "giraffe".

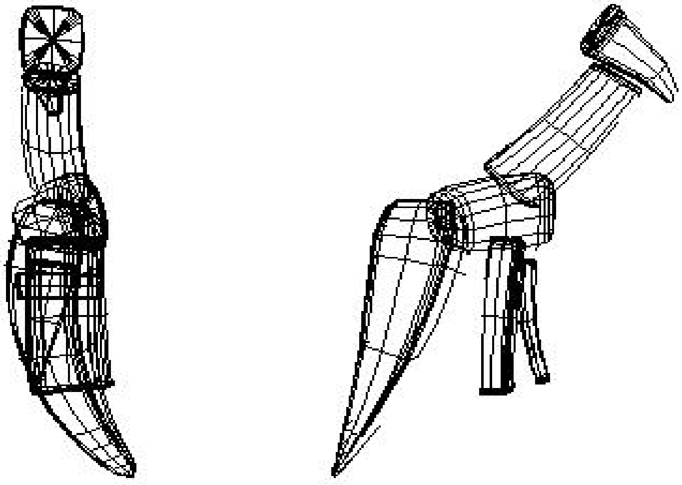


Figure 3.24: Superquadrics recovered from data object "giraffe" shown in two different positions.

account for analyzing the performance of segmentation.

Appendix F includes thirteen (13) more range images tested for the recognition approach. Seven (7) different objects are shown: Figures (F.1 (a), F.1 (b), F.2 (a), F.2 (b), F.3) are range images of objects "bear", "elephant", "cheetah", "hippopotamus", and "kangaroo" respectively; Figures (F.4 (a), F.4 (b), F.5 (a), F.5 (b)) are of object "horse" (actual object "horse" in Appendix F is different from the one shown previously in this chapter) in four (4) different views; and Figures (F.6 (a), F.6 (b), F.7 (a), F.7 (b)) are of object "pig" also in four (4) different views.

Figures (F.8 to F.14) show the discontinuities found on the thirteen (F.1 to F.7) original range data, and Figures (F.15 to F.27) show the closed regions as resulted from segmentation.

Together with the results given previously in this chapter, the set in Appendix F provide data upon which to discuss some issues of stability in the volumetric segmentation as a whole. Analyzing Figures (F.8 to F.14) the discontinuities were detected where expected in all the images, the effects of changing the viewing positions are noticed

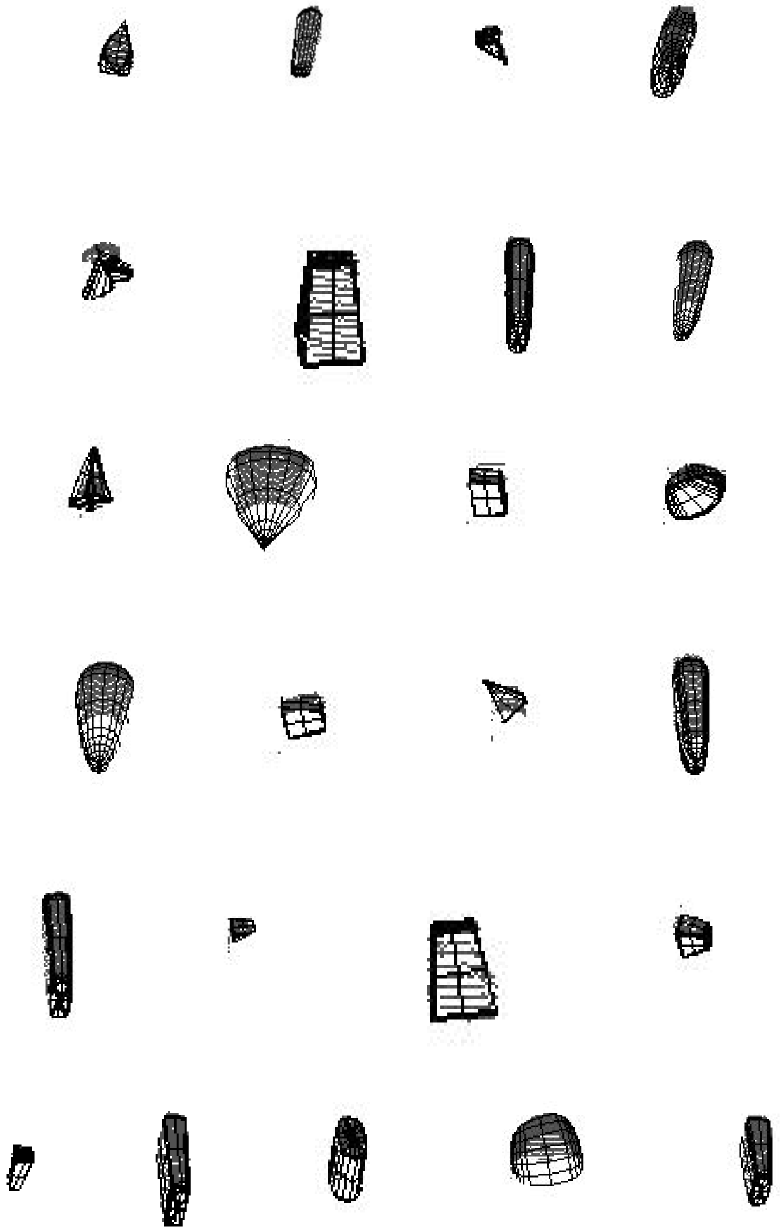


Figure 3.25: Superquadrics wire-frames shown on top of each region separately for object “doll”.

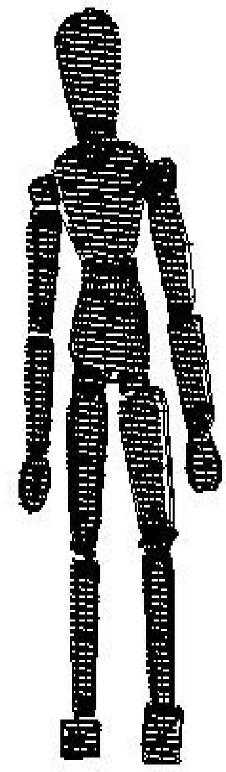


Figure 3.26: Superquadrics wire-frames shown on top of fitted regions for object “doll”.

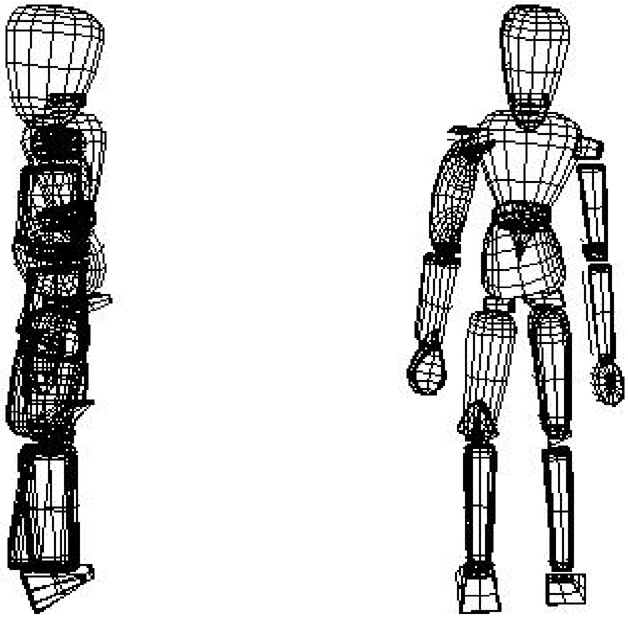


Figure 3.27: Superquadrics recovered from data object "doll" shown in two different positions.

more significantly for the object "pig" (Figures (F.13 (a), F.13 (b), F.14 (a), F.14 (b))). The discontinuities detected for object "horse" (Figures (F.11 (a), F.11 (b), F.12 (a), F.12 (b))) kept some uniformity and did not make much difference for the result of segmentation. For object "cheetah" (Figure (F.9 (a))) the parts of the "backleg" and the "tail" contain a large number of discontinuities, relatively to the size of the expected part. The outcome of this in the segmentation was that the recovery of the superquadric parameters for these parts were not successful. In fact, as can be seen from Figure (F.29 (a)) the parts of the "backleg" and "tail" were not matched successfully.

The other object for which segmentation did perform badly was the "pig" (Figures (F.13 (a), F.13 (b), F.14 (a), F.14 (b))). As can be seen from the Figures (F.24, F.25, F.26, F.27), which show the closed regions for object "pig", the number of regions found for the view in Figure (F.24) is different from the others, having the parts relative to the legs being joined with the body.

Objects	regions extracted	regions modelled	regions missing	unexpected regions
horse (first)	9	9	0	0
cow	8	8	0	0
giraffe	7	7	1	1
doll	25	25	0	0
bear	6	7	1	0
elephant	8	8	0	0
cheetah	8	8	0	0
hippopotamus	5	7	2	0
kangaroo	7	8	1	0
horse (view 1)	9	8	0	1
horse (view 2)	8	8	0	0
horse (view 3)	7	8	1	0
horse (view 4)	9	8	0	1
pig (view 1)	6	9	3	2
pig (view 2)	9	9	0	0
pig (view 3)	9	9	0	0
pig (view 4)	8	9	1	0

Table 3.1: Summary of the number of regions found by the segmentation for all seventeen images tested, together with the number of expected and unexpected regions.

From this analysis it can be said that the discontinuity closing module of the segmentation performed successfully by finding the expected closed regions, even for more difficult cases when parts had many discontinuities (Figure (F.9 (a)), and when objects with small parts had a large number of undetected discontinuities considering the different views (Figures (F.13 (a), F.13 (b), F.14 (a), F.14 (b))).

Table 3.1 summarizes the number of regions segmented for all the seventeen range images tested, the number of regions from the respective models, and the numbers of missing and unexpected regions found. From the data shown in Table 3.1 it can be seen that for $rm = 155$ regions modelled, for all the objects, $rmi = 10$ were missed by the segmentation, which gives an overall figure of 93.6% of the expected regions found. From the $re = 148$ regions extracted a number $rne = 5$, which represents 3.4%, of unexpected regions were extracted. These results show that for most conditions tested the first part of the segmentation is stable.

The stage of the segmentation that is more fragile to these perturbations is the superquadric recovery process. The superquadric estimation procedure depends strongly on the number of 3-D points of a part which is being fitted, and on the view from which the surface is being scanned. From the data tested (including the ones shown in this chapter and Appendix F), the superquadric parameters estimation has shown bigger variations, and in consequence worse performance: for the “backleg” and “tail” parts of object “cheetah”, because of the large amount of discontinuities in these parts; and for almost all the parts of object “pig”, mainly because of the small size of parts (the actual size of object “pig” is approximately three times smaller than the others).

The stability of the segmentation approach presented here can be appraised, considering also the results shown in Appendix F, by observing that even for more difficult data (smaller objects, different views) the general form of the estimated superquadrics was correctly extracted. An important result of this analysis is to notice the effect on the superquadric estimation of introducing small perturbations on the range data of objects (*e.g.* different views). These perturbations have only affected the quantitative, *i.e.* numerical values, of the superquadrics parameters.

The effect on the performance, of the superquadric recovery procedure under changing conditions, can be seen more quantitatively by looking at the results from Table 3.1 for objects “horse” (views 1-4) and “pig” (views 1-4). Although the difference between the number of regions found and the number of regions modelled was for the worst case (“pig” view 1) 33.33%, and on average it was around 14.0% for these two objects. The object “pig” was not successfully matched (see more results given in the next chapters) mainly because of inaccurate superquadric parameters estimated from the data. For object “horse” (views 1-4), although the changing viewing positions perturbed also the superquadric estimation procedure, the results were less affected because the 3-D regions of object “horse” are more elongated, bigger and smoother than the pig’s regions.

As seen some perturbations affect more heavily the estimation of the superquadric parameters than others. This fact makes stronger the importance of the two next stages of the recognition approach, (*i.e.* the geon classification and the fuzzy matching), because the next stages will have to account for this variation in the parameters of the superquadrics and solve for acceptable matches.

3.8 Chapter Summary

Parts-based segmentation of 3-D articulated objects is an important topic and, as yet, relatively unexplored in the literature. In order to provide as stable and accurate a solution as possible, one has to consider recovering and grouping the discontinuities on the surface of the object, as well as representing the segmented pieces in a way that is visually appealing, easy to extract and powerful for indexing. The combination of a discontinuity preserving curvature estimation technique, a dynamic grouping of the main discontinuities involved, and a fitting of deformable volumetric models of superquadrics represents one good way of approaching this problem.

Segmentation of 3-D objects into prototypical parts provides a vision system with the possibility of reasoning about the location, shape and function of complex natural and man-made objects with articulations. The differential structure of the object's surface gives useful information regarding where an articulated object might be segmented, as it was shown by our results in this Chapter. The mixed approach presented here produces a stable estimate of the local structure of the surface without smoothing across discontinuities. This allows a characterization of the local structure more precisely than simple curvature consistency alone. The use of globally deformable superquadrics, as a representation for the segmented parts, is especially appealing because of their great capability to capture the notion of 3-D parts in natural form.

One drawback of this segmentation approach is that it is time consuming. For example, an image like the one in Figure 3.7 (256x256) which was segmented in 25 regions can take up to 2 hours of CPU time on a SUN SparcStation 2 considering all the stages of the segmentation. The computation of smoothed principal directions for the 3-D surface, and the dynamic contours linking are the most expensive tasks in the approach. One possible solution to this would be to investigate discontinuities at different scales and devise a heuristic approach to link them. Another method would be to hypothesize a large number of regions initially, by first an analysis of the discontinuities and the types of surface patches found, and then placing different fittings of the superquadrics which will include a "winner-takes-all" (*e.g.* as in [Darrell *et al.* 90]) type of decision to choose the most appropriate set of volumes in the end. A different approach would be to consider multiple view data and to acquire the volumetric parts by taking into account the segmentation in each view.

Segmentation of complex 3-D objects such as those dealt with in this thesis into volumetric primitives is a difficult task. In this chapter we have presented a suitable way to tackle this problem which contributed to previous work done in the area in the following aspects:

- The segmentation procedure developed here extracts globally deformable superquadrics from 3-D curved objects for the purpose of model-based recognition.
- The computation of the differential structure of the object surface (i.e. accurate Darboux Frames) utilizes a pre-processed discontinuity map to guide the quadratic fitting and to avoid smoothing it in the process.
- Tapering and bending deformations are included in the superquadrics fitting in order to provide a bigger set of primitives which satisfies the primary objective of recognition of natural forms.

Chapter 4 presents a method to classify superquadric primitives segmented in this chapter into geons in order to add qualitative properties to the representation and deliver them to the matching stage later.

Chapter 4

A Radial Basis Function Neural Net for Classification of Superquadrics into Geons

This chapter describes a method of classifying superquadrics with global deformations into a set of qualitative volumetric primitives. The superquadrics are obtained by a non-linear least squares fit of segmented regions as described in Chapter 3. Superquadric shapes are characterized by the continuum of their parameters, which can present a difficult problem if one wants to use them for matching. The solution is to use ranges of the parameters to identify a particular class of shape. For the purpose of model-based matching the representation includes both quantitative terms, the superquadric parameters, and qualitative terms, the label of the class. As will be shown in the next chapters, it is suitable for model-based recognition.

The set of qualitative volumetric shapes into which the superquadrics are classified is a subset of the **geons** [Biederman 87], which are representative of a large variety of 3-D objects found in natural and man-made environments.

The method presented in this chapter for the classification of superquadrics into geons is a new contribution to this problem in Computer Vision. The following sections present the problem and our solution, and analyze related work.

4.1 Choosing a Set of Qualitative Volumetric Shapes: From Superquadrics to Geons

Using the superquadric parameters for 3-D object recognition directly would not be very efficient because they lack expressiveness for indexing into the object database. This problem can be solved by mapping the superquadric parameters into a set of distinctive volumetric shapes with good potential for indexing. In the end, both qualitative and quantitative shape information would be readily available.

One set of distinctive volumetric shapes are the **geons**, which were proposed as part of the Recognition-by-Components theory of Biederman [Biederman 87]. In his theory, Biederman proposes that a set of 36 fundamental volumetric part primitives (geons) can represent many complex 3-D objects for the purpose of primal access. The Recognition-by-Components theory builds on analyzes of perceptual organization, especially the work of Lowe [Lowe 85], which identifies nonaccidental geometric relations on visual forms in order to infer a small set of volumetric primitives. These nonaccidental relations are:

1. Colinearity of points and lines.
2. Curvilinearity of points of arcs.
3. Symmetry.
4. Parallel curves.
5. Vertices.

Applying different variations in nonaccidental relations of four attributes of generalized cylinders Biederman arrived at a set of 36 basic components, which constitute the geons. Figure 4.1 (adapted from [Biederman 87]) shows an illustration of how variations in attributes of the cross-section (edge, size, and symmetry) and the shape of the axis of a cylinder can generate a subset of the family of geons.

The four attributes or geometrical properties Biederman used to define the 36 geons, namely cross-section edge, cross-section size, cross-section symmetry, and axis shape

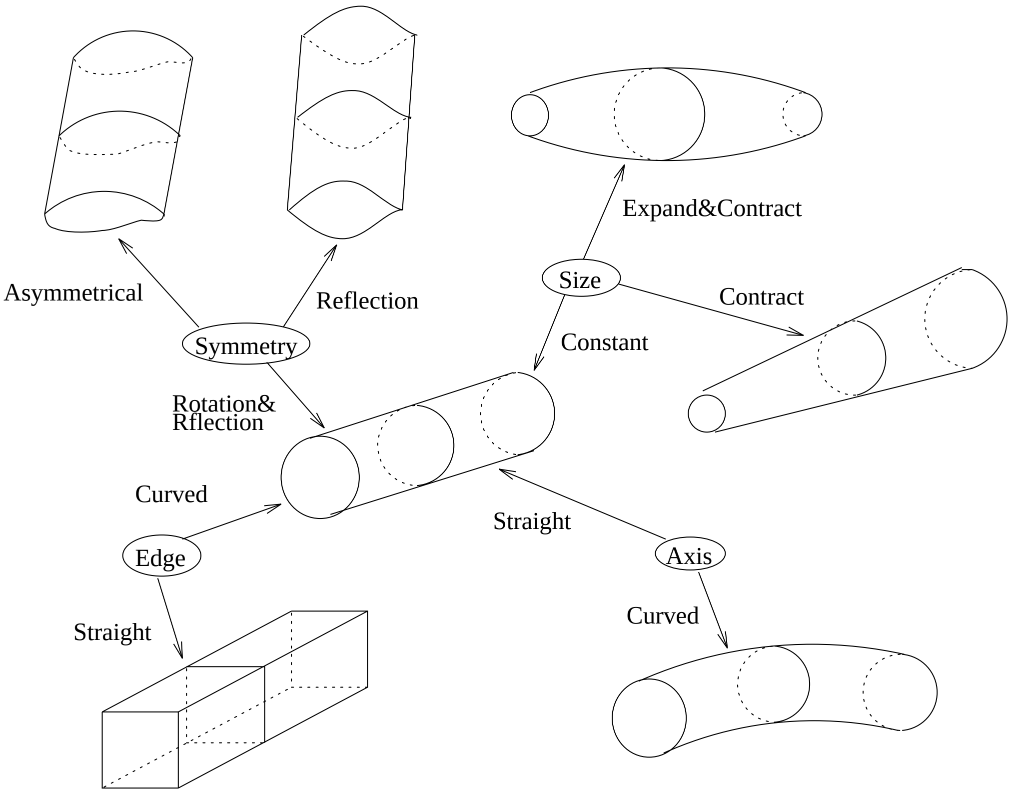


Figure 4.1: Illustration of how variations in attributes of a cylinder generate a set of generalized cylinders differing from nonaccidental properties (adapted from [Biederman 1987]).

are not always considered useful or important in achieving shape discrimination. This is particularly true for the case of the cross-section symmetry attribute, because even though it generates different volumetric primitives, it is not commonly found in many objects, and the detection of this property is not straightforward. Other works that also use geons [Dickinson *et al.* 92a, Raja & Jain 92] do not consider cross-section symmetry and deal with a subset of the original 36 geons.

With the set of deformations used here in the superquadric parameters (tapering and bending) 12 geon classes can be modelled. As discussed above the cross-section symmetry attribute is expensive to detect and the other three attributes are sufficient for most practical situations. The input to the classifier is segmented data (superquadric descriptions), and what is important is having enough indexing capability (qualitative shape information) to improve our representation. Figure 4.2 shows the 12 geon classes used as the target mapping set.

The labels in Figure 4.2 indicate the following sequence of the three geometrical attributes:

- The first letter **-s-** or **-b-** relates to the type of axis shape, if straight or bent.
- The second set of letters indicates the type of cross-section edge, either **-c-** curved or **-s-** straight.
- The last set of letters indicates the cross-section size, if it is **-co-** constant, **-id-** increasing-and-decreasing, or **-ta-** tapered.

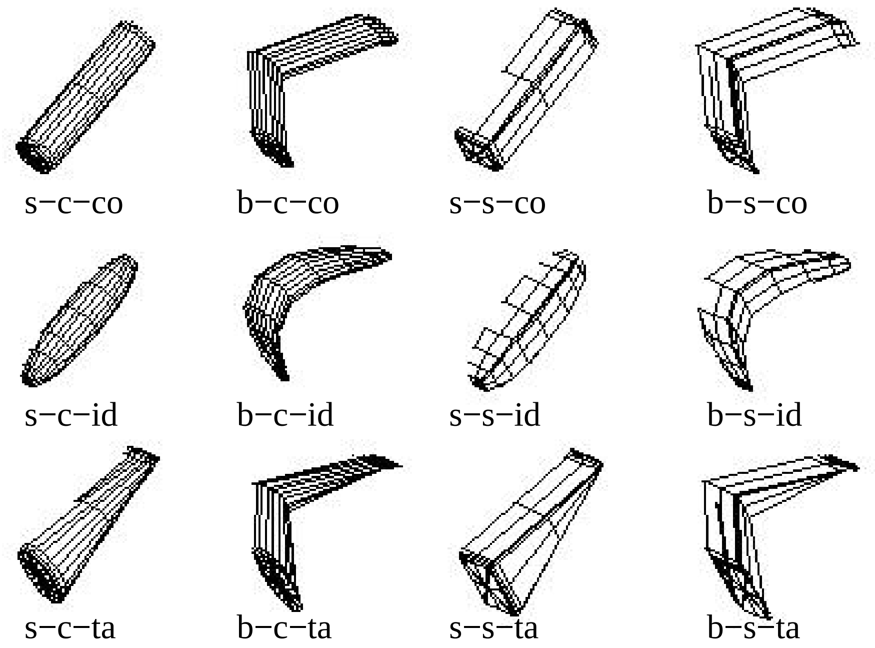


Figure 4.2: The set of twelve (12) geons modelled using the attributes of axis shape, cross-section edge, and cross-section size.

This mapping from superquadrics to geons does not depend on all 15 recovered superquadric parameters because only 9 of them are related to shape, the other 6 give the position and orientation of the superquadric. From these 9 parameters three are the elongations along the axes a_1, a_2, and a_3, two are the shape parameters ϵ_1, and ϵ_2, two are the tapering along x and y, T_x and T_y, and two are related to bending, the radius r and the angle β. The bending property can be represented by using a combination of the radius of bending r and the parameter a_3 [Raja & Jain 92], as

$$rad\_bend = r/a_3 \tag{4.1}$$

The fitting procedure of the superquadrics chooses as convention the longest axis to be the z axis, therefore a_3 represents the longest elongation of the object and equation 4.1 gives a clear representation of the bending.

Therefore, in order to recognize the 12 geons shown in Figure 4.2 we need only the following 5 superquadric parameters:

- ϵ_1.
- ϵ_2.
- T_x, tapering in x.
- T_y, tapering in y.
- rad_bend.

The classification task is to map the 5 superquadric parameters given above into 12 distinct classes of geons. Reconstructing such complicated mapping functions as efficiently as possible (*i.e.* achieving high accuracy using small amounts of time and training data) is an active area of research in neural networks [Lippmann 89]. Section 4.2 introduces a Radial Basis Function network which achieves higher classification rates than other previously published algorithms for the problem at hand, making it possible in our case to deduce more accurate qualitative information for visual matching of three dimensional shapes.

4.2 A Radial Basis Function Neural Net Classifier

The method of Radial Basis Functions (RBF) is a mathematical technique for solving interpolation problems in high dimensional spaces. Recent developments and reviews can be found in [Powell 87, Jackson 89]. The basic interpolation problem can be stated as follows:

Basic Interpolation Problem. Given a set x_k of data points such as

$$\{x_k \in \Re^p; k = 1, 2, ..., n\} \tag{4.2}$$

and function values

$$\{y_k \in \Re^q; k = 1, 2, ..., n\} \tag{4.3}$$

find a function $f : \Re^p \rightarrow \Re^q$ that satisfies

$$f(x_k) = y_k, \qquad k = 1, 2, ..., n. \tag{4.4}$$

In a straightforward regression problem the normal procedure is to assume that the data is explained by

$$y \;=\; f(x;\, W) + \epsilon \;, \tag{4.5}$$

where $f(\cdot;\, \cdot)$ is a function which depends on some unknown parameters, W, and ϵ is zero mean additive noise. The unknown parameters can be estimated by some optimization criterion such as least squares. For classification problems the same technique often works in practice. In that case the components of the training outputs $\{y_k\}_1^n$ are all zero except the component which corresponds to the correct output class, which is unity. Then the regression function $f(x;\, W)$ estimates the probabilities of each of the q classes given the input (features) x.

RBF networks assume that the components of the vector function $f(\cdot,\, \cdot)$ are given by

$$f_i(x;\, W) \;=\; \sum_{j=1}^{m} W_{ji}\, h(x,\, c_j) \;, \quad 1 \le i \le q \;, \tag{4.6}$$

a sum of functions which depend on m fixed positions (centres), $c_j \in \Re^p$, in the input space and an m-by-q weight matrix whose rows are points in the output space. Such a function can be viewed as a network (see Figure 4.3). Certain conditions must be met by the function $h(\cdot,\, \cdot)$ such as being radial and monotonic for positive values. We use the Gaussian function

$$h(x,\, c) = \exp\left(-\frac{\| x - c \|^2}{r^2}\right) \ , \tag{4.7}$$

where the scaling parameter r is, like the $\{c_j\}_1^m$, chosen in advance. As the function is linear in the unknown parameters (W), the least squares criterion leads to a system of linear simultaneous equations to solve, a less daunting prospect than the nonlinear optimization problems posed by some other types of neural networks.

A feed-forward layered network model of Radial Basis Functions was first introduced by Broomhead and Lowe [Broomhead & Lowe 88]. In this model the solution is organized in a neural network style as the one seen in Figure 4.3.

4.3 Solving the Reconstruction Problem by Regularization

One of the advantages of the RBF neural networks over other frameworks for classification is that the learning phase requires solving a linear problem, even though a nonlinear mapping can be achieved.

A common problem with all regression and classification methods is the trade off between bias and variance [Geman *et al.* 92]. If the model is too flexible it will overfit (too much variance) and if it is not flexible enough it will underfit (too much bias). In RBFs there are basically three frameworks to deal with this:

1. Regularization (*e.g.* [Bishop 91]);
2. Forward centre selection (*e.g.* [Chen *et al.* 91]) for building small networks until fitting, and
3. Randomly selects a number of centers (*e.g.* [Broomhead & Lowe 88]) from the training data.

In this thesis the regularization framework, which stems from the work of Tikhonov and others [Tikhonov & Arsenin 77], is used and a new algorithm is designed for the

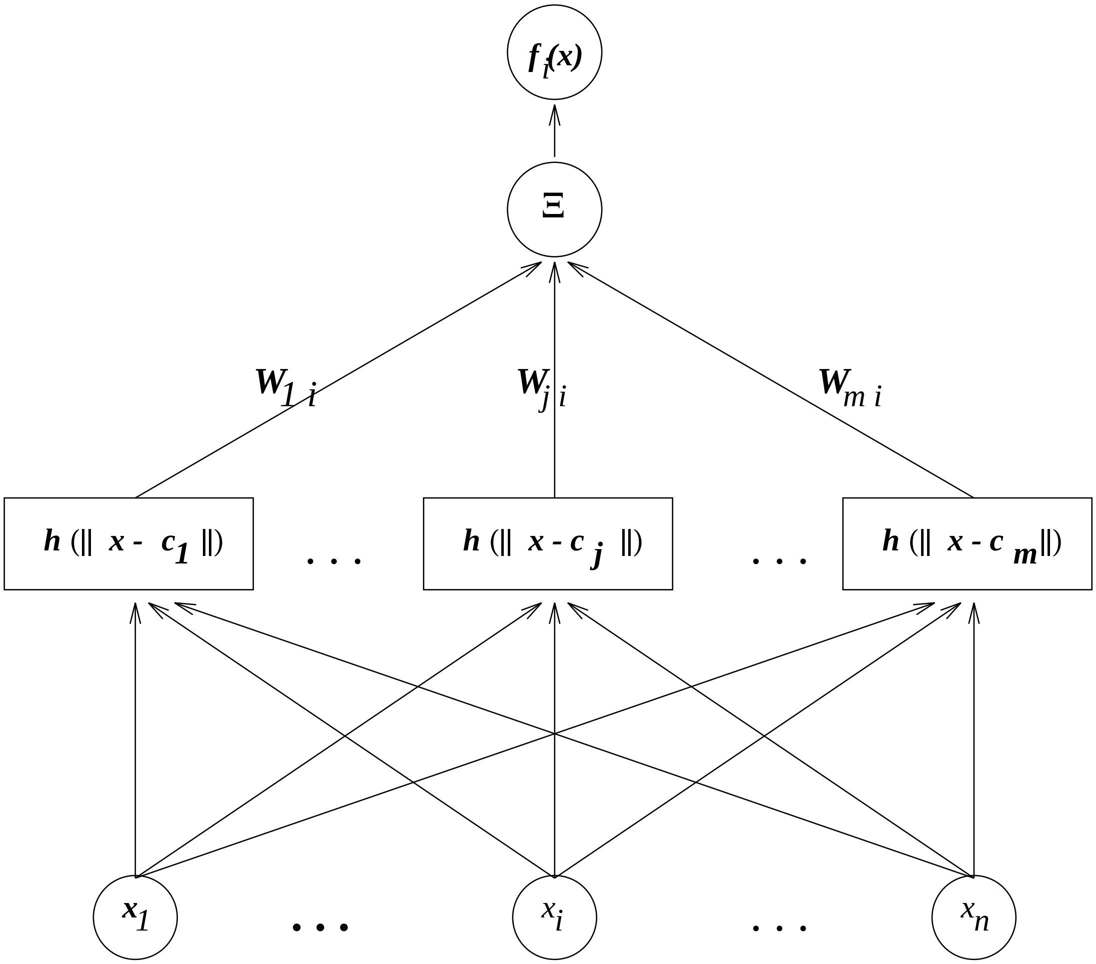


Figure 4.3: Radial Basis Function Neural Network.

problem of classifying superquadric parameters into a set of qualitative volumetric shapes.

A common choice for the fixed centres are the n input points in the training data, $\{x_k\}_1^n$ so that $m = n$ and $c_j = x_j$. Then the normal solution to the least squares minimization is

$$W = (H^T H)^{-1} H^T Y \ , \tag{4.8}$$

where

$$H_{kj} = h(x_k, x_j) \ , \quad 1 \le k, j \le n \ , \tag{4.9}$$

and the rows of the n-by-q matrix Y are the training set outputs. The problem with this choice of centres is that the number of parameters to estimate is the same as the number of data available and any redundancy in the data will lead to overfit. Often also the matrix $H^T H$ will be ill-conditioned. The simplest regularization technique to avoid such overfit (and also ill-conditioning) is to add a weight penalty term to the least squares criterion so that the function to be minimized (with respect to W) is

$$\sum_{k=1}^{n} \| y_k - f(x_k, W) \|^2 + \lambda \sum_{j=1}^{m} \sum_{i=1}^{q} W_{ji}^2 \, , \tag{4.10}$$

(instead of just the first term). This is called zero-order regularization and λ is a positive constant called the regularization parameter [Press *et al.* 92]. The solution to this minimization problem is

$$W \;=\; (H^T H + \lambda I)^{-1} H^T Y \, . \tag{4.11}$$

The idea is that a small $\lambda > 0$ will introduce a small bias but greatly reduce the variance.

We now have two free parameters in the RBF, the Gaussian radius r and the regularization parameter λ. The optimal values to use for these parameters depend on the amount and characteristics of the training data available for the classifier. However, there exist criteria for ranking different classifiers on the same data and one of these, generalized cross-validation (GCV) [Golub *et al.* 79], is employed here. The GCV score is given by

$$\mathrm{GCV} = \frac{\mathrm{trace}(Y^T P^2 Y)}{[(1/n)\,\mathrm{trace}(P)]^2} \, , \tag{4.12}$$

where

$$P = I - H\,(H^T H + \lambda I)^{-1} H^T \, , \tag{4.13}$$

and lower GCV scores are associated with better generalization.

Figure 4.4 shows a contour plot of GCV (in arbitrary units) for different choices of r and λ for the superquadric to geon classification data (see next section). There is a clear minimum near the point $r = 7$, $\lambda = 10^{-6}$, and the classification results we describe below are based on these values.

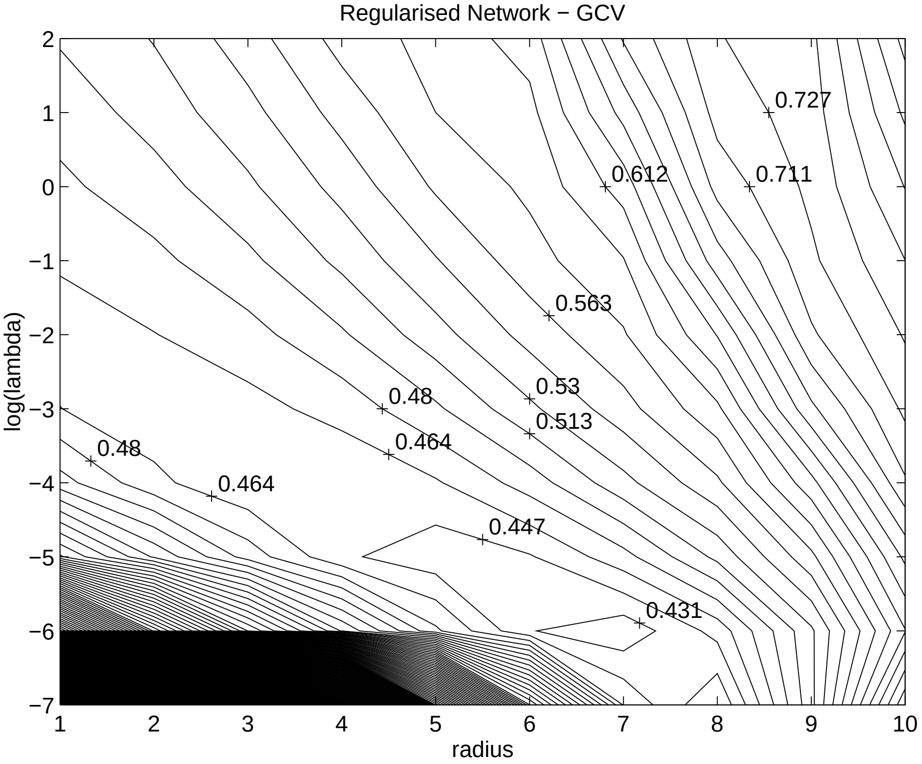


Figure 4.4: Regularized RBF Neural Network using Cross-Validation

4.4 Experiments and Results

Data used for training the classifier was acquired by the segmentation procedure for range images using the algorithm presented in Chapter 3, as applied to a set of wooden and plastic articulated objects. Previous results of the application of RBF classifier were presented by us in [Borges *et al.* 94].

Training of the RBF classifier is done off-line with the final matrix of learned weights W being stored in a function for classification of new data. The training method adopted was a supervised procedure where initially twelve ($q = 12$) prototypes were chosen to signal the classifier its error. 369 examples were presented to the classifier, testing was performed using a hold-out method (with 10% $= 37$ held out and $n = m = 332$ randomly selected training samples). The final parameters were considered stable after 40 trials of experiments, when changes in the parameters were very minimal.

Proper values for the $p = 5$ superquadric parameters ($\epsilon_1, \epsilon_2, t_x, t_y,$ and rad_bend)

	ϵ_1	ϵ_2	T_x	T_y	rad_bend
b-c-co	≤ 0.3	> 0.3	≤ 0.3	≤ 0.3	≤ 6.5
b-c-id	≥ 0.7	> 0.3	≤ 0.3	≤ 0.3	≤ 6.5
b-c-ta	≤ 0.3	> 0.3	> 0.3	> 0.3	≤ 6.5
b-s-co	≤ 0.3	≤ 0.3	≤ 0.3	≤ 0.3	≤ 6.5
b-s-id	≥ 0.7	≤ 0.3	≤ 0.3	≤ 0.3	≤ 6.5
b-s-ta	≤ 0.3	≤ 0.3	> 0.3	> 0.3	≤ 6.5
s-c-co	≤ 0.3	> 0.3	≤ 0.3	≤ 0.3	> 6.5
s-c-id	≥ 0.7	> 0.3	≤ 0.3	≤ 0.3	> 6.5
s-c-ta	≤ 0.3	> 0.3	> 0.3	> 0.3	> 6.5
s-s-co	≤ 0.3	≤ 0.3	≤ 0.3	≤ 0.3	> 6.5
s-s-id	≥ 0.7	≤ 0.3	≤ 0.3	≤ 0.3	> 6.5
s-s-ta	≤ 0.3	≤ 0.3	> 0.3	> 0.3	> 6.5

Table 4.1: Range of values expected for clear examples of the 12 geons.

identified as the C_j input features to discriminate between the 12 geons are chosen by experience, *i.e.* verifying by means of a modelling system like THINGWORLD [Pentland & Sclaroff 91] which values should be expected for clear examples of the 12 classes as shown in Figure 4.2. These values are shown in Table 4.1 and are in agreement with those of [Raja & Jain 92], where other algorithms were applied to a similar classification problem. Their approach is analyzed further in this chapter.

In order to achieve the results shown in Tables 4.2, 4.3, and 4.4, $n = 369$ input-output training examples were used, including sometimes the same object to be segmented but in different positions. We included approximately the same number of examples ($\simeq 30$) from each of the 12 classes.

Table 4.2 gives the mean and standard deviations of the 5 parameters used in the experiments for the 12 geon classes. Table 4.3 shows the success rates for the classifier using the hold-out method (which is known to give a pessimistic estimate) indicating the individual and the overall rates for all the classes. The method involves training the classifier on some percentage of the data (we used 90%) and testing it on the rest which is held in reserve. The process is repeated a number of times (40) to get an average success rate.

Table 4.4 shows the confusion matrix accumulated over 40 runs of the RBF classifier. Entry (i, j) in Table 4.4 means the percentage where the true class was i and

Geon type	ϵ_1		ϵ_2		t_x		t_y		rad_bend	
	mean	st.dv.	mean	st.dv.	mean	st.dv.	mean	st.dv.	mean	st.dv.
b-c-co	0.135	0.110	0.694	0.227	0.180	0.129	0.190	0.131	3.588	1.833
b-c-id	0.771	0.192	0.776	0.174	0.388	0.305	0.310	0.251	3.607	1.606
b-c-ta	0.157	0.109	0.693	0.245	0.690	0.226	0.512	0.299	3.846	1.677
b-s-co	0.161	0.094	0.201	0.092	0.176	0.123	0.128	0.121	4.133	1.714
b-s-id	0.766	0.242	0.218	0.243	0.397	0.309	0.337	0.327	4.275	1.623
b-s-ta	0.083	0.097	0.153	0.088	0.658	0.309	0.605	0.297	4.270	1.378
s-c-co	0.139	0.107	0.722	0.210	0.140	0.122	0.158	0.160	16.037	15.714
s-c-id	0.752	0.195	0.758	0.172	0.412	0.307	0.346	0.310	15.406	9.648
s-c-ta	0.153	0.144	0.682	0.179	0.613	0.223	0.614	0.250	16.198	14.616
s-s-co	0.132	0.108	0.131	0.105	0.132	0.100	0.119	0.102	11.701	8.361
s-s-id	0.756	0.200	0.150	0.106	0.302	0.312	0.306	0.308	18.668	20.332
s-s-ta	0.089	0.090	0.160	0.083	0.627	0.304	0.631	0.326	15.983	11.877

Table 4.2: Mean and Standard Deviation Values for the Classification Parameters of each Class of Geon

the estimated class was j. It can be seen from this confusion matrix that the main false classifications occur along two minor diagonals, {(7,1), (8,2), (9,3), (10,4), (11,5), (12,6)} and {1,7), (2,8), (3,9), (4,10), (5,11), (6,12)}. The false classifications along these two minor diagonals happen with confusion of the axis shape attribute, being either bent or straight, and occur because of the presence of slightly bent shapes whose properties are near the decision boundary. Using more training data would certainly refine the decision boundary and eliminate some of these false classifications, however the positive classification rate already achieved in our experiments is high and well acceptable for our purposes. It will be an issue for future investigation how much improvement could be made with the acquisition of more training data, and whether other tuning algorithms could be developed. Meanwhile, the rates achieved in these experiments show the RBF classifier presented here is a well tuned solution to our problem.

Figures 4.5, 4.6, 4.7, 4.8 show the 3-D original data for the objects "horse", "cow", "giraffe", and "doll" with the fitted superquadric primitives overlayed and labelled according to the best classification from the RBF classifier.

Altogether seventeen (17) range images were used and thirteen (13) were successfully recognized for a total rate of 76%.

Geon type	Hold-out success %, 40 averages, including only the highest classification	Hold-out success %, 40 averages, including the three highest classifications
b-c-co	75.0	92.4
b-c-id	71.6	96.6
b-c-ta	92.8	100.0
b-s-co	64.0	87.1
b-s-id	74.8	92.7
b-s-ta	74.3	94.5
s-c-co	88.4	97.3
s-c-id	88.6	98.9
s-c-ta	81.0	94.8
s-s-co	81.5	92.3
s-s-id	85.3	96.1
s-s-ta	78.1	94.8
Overall (all classes)	79.0	94.4

Table 4.3: Rates of Classification for the RBF Neural Net Classifier

	b c co	**b c id**	**b c ta**	**b s co**	**b s id**	**b s ta**	**s c co**	**s c id**	**s c ta**	**s s co**	**s s id**	**s s ta**
b-c-co	75.0	0.0	0.0	11.5	0.0	0.0	5.5	0.0	0.0	0.0	0.0	0.0
b-c-id	4.5	71.6	0.0	0.0	6.5	0.0	0.0	4.5	0.7	0.0	1.0	0.0
b-c-ta	12.1	0.0	92.8	0.7	0.0	7.3	0.0	0.0	6.5	0.0	0.0	0.0
b-s-co	1.5	0.0	0.0	64.0	2.4	0.0	0.0	0.0	0.0	5.4	0.0	0.0
b-s-id	5.3	11.2	0.0	5.8	74.8	5.5	0.0	0.0	0.0	1.5	3.9	0.0
b-s-ta	0.0	0.0	2.4	6.5	4.9	74.3	0.0	0.0	0.0	0.0	0.0	3.1
s-c-co	1.5	0.0	0.0	0.0	0.0	0.0	88.4	0.0	7.2	2.3	0.0	0.0
s-c-id	0.0	13.8	0.0	0.0	0.0	0.0	2.1	88.6	2.6	0.0	6.9	0.0
s-c-ta	0.0	3.4	4.8	0.0	0.0	0.0	3.4	5.7	81.0	0.0	0.0	10.4
s-s-co	0.0	0.0	0.0	11.5	0.0	0.0	0.0	0.0	0.0	81.5	0.0	2.1
s-s-id	0.0	0.0	0.0	0.0	11.4	0.0	0.7	0.0	0.0	3.8	85.3	6.3
s-s-ta	0.0	0.0	0.0	0.0	0.0	12.8	0.0	1.1	2.0	5.4	2.9	78.1

Table 4.4: Confusion matrix accumulated over 40 runs of the RBF classifier. Entry (i, j) in the table means the percentage % where the true class was i and the estimated class was j.

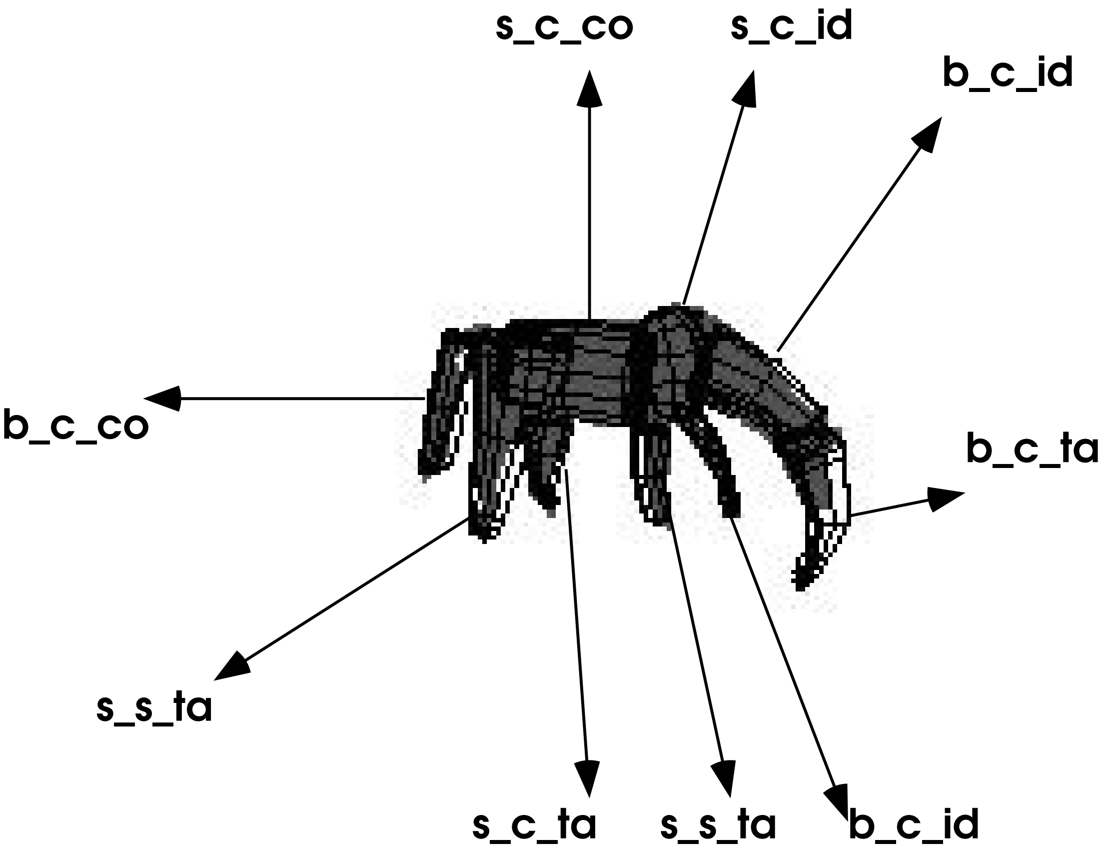


Figure 4.5: Fitted superquadric primitives shown overlayed on original 3-D data for object "horse" with geons labels from RBF classification.

4.5 Discussion and Criticisms

The main purposes of the geon classification stage for the recognition approach developed in this work are twofold: first, it is to provide a symbolic index for the volumetric features of the objects' parts with which a more efficient search could be designed; and second, it is to accommodate the range of numeric values, on which the superquadric shape parameters vary, on prototypical volumes that are representative of that class of shapes.

First evaluation can be focussed on the RBF classifier algorithm itself and its rates of success for the superquadrics-into-geons classification. As shown in Table 4.3 the success rates achieved for the training performed had an overall score of (79.0%). This number is a high score considering this type of task, although it is a more subjective

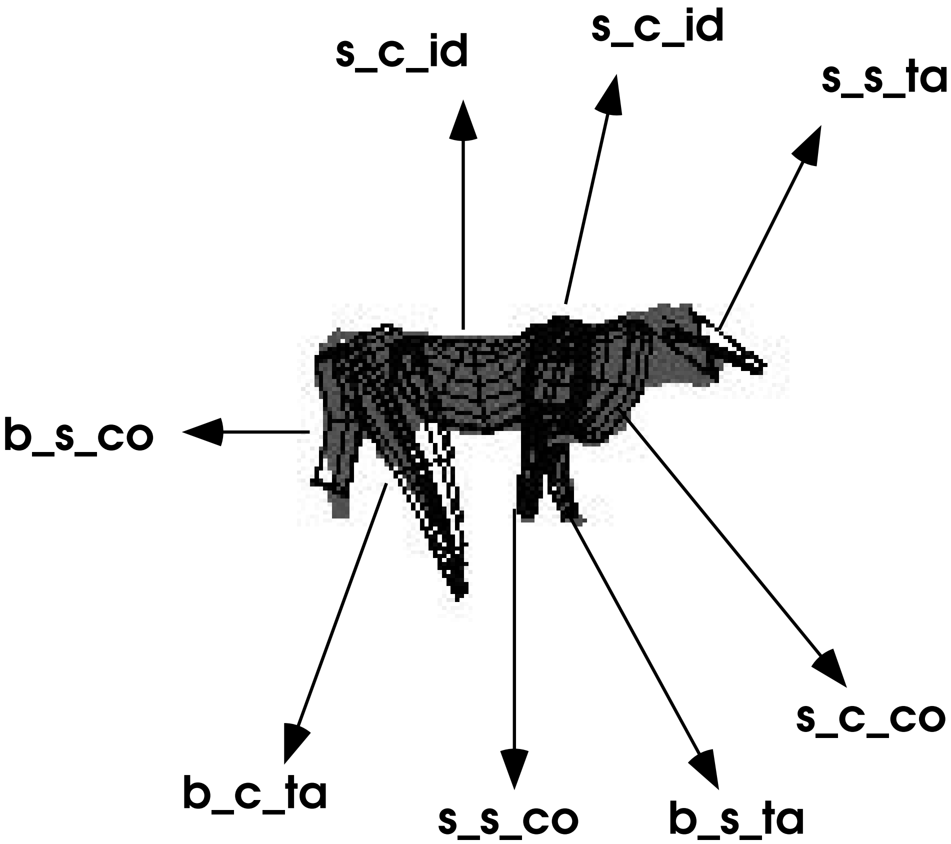


Figure 4.6: Fitted superquadric primitives shown overlayed on original 3-D data for object "cow" with geons labels from RBF classification.

judgement of performance since we do not have results on the same experimental setup for other algorithms. By looking at the individual success rates achieved for each geon type, the lowest rate was for geon type **b-s-co** (64.0%) (second column, Table 4.3). This was an expected measure since the shape of this geon type is less distinguishable from the superquadric parameters than the eleven other geons.

Table 4.5 summarizes the classifications performed by the RBF classifier over the seventeen images tested in this thesis. The figures of correct and incorrect classifications were derived by comparing the result of the classification (output of RBF), directly to the labels from the respective models. For a total number of $cg = 148$ classifiable geons, $cc = 107$ were classified as expected (72.3%), and $ic = 41$ were badly classified (27.7%). If the same figures are taken only for the thirteen images that were successfully matched, the numbers are ($cg = 116; cc = 93; ic = 23$), which give a success rate

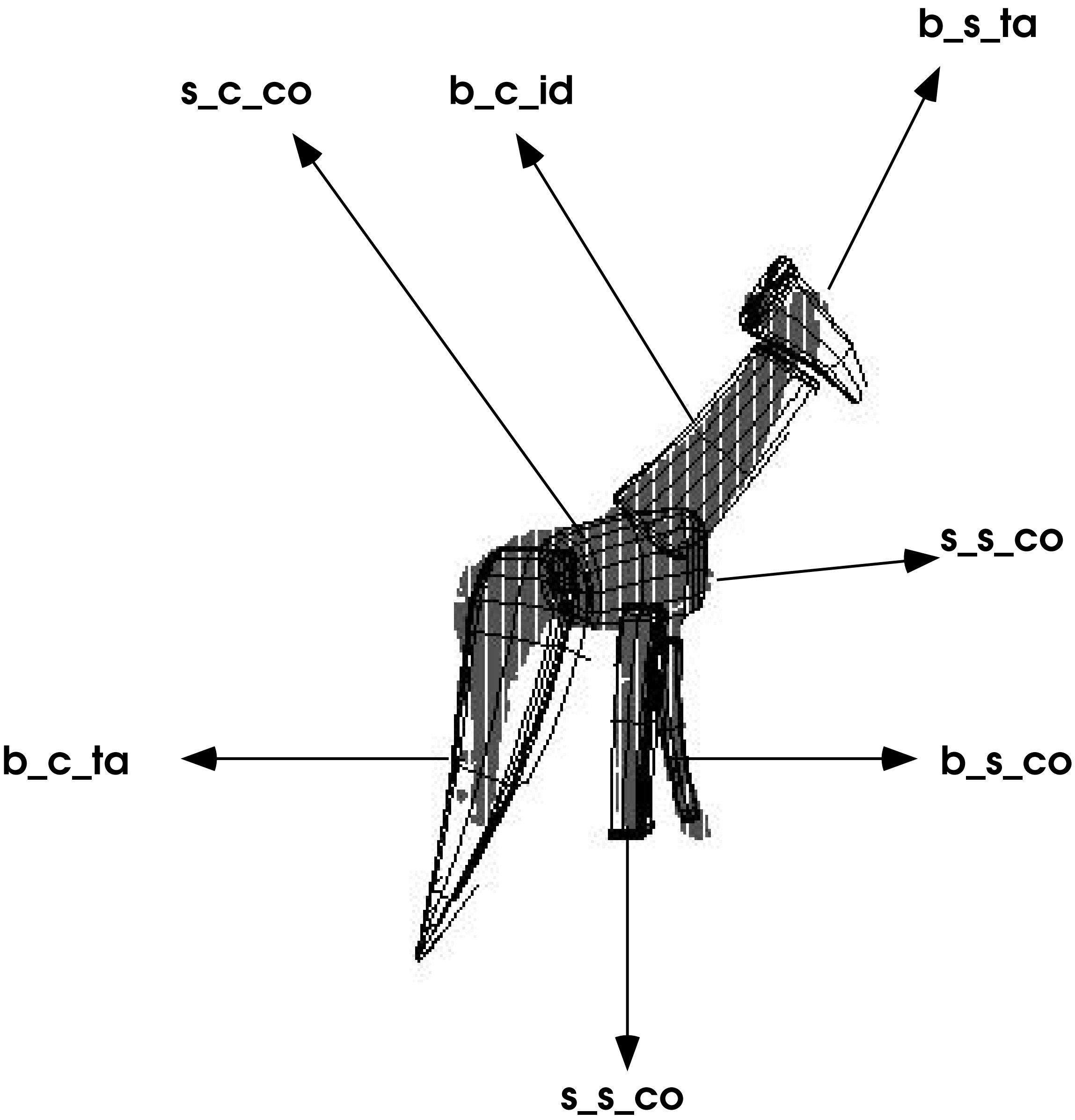


Figure 4.7: Fitted superquadric primitives shown overlayed on original 3-D data for object "giraffe" with geons labels from RBF classification.

of 80.2%.

Another point to discuss is the value of this geon classification for providing evidence that could be used in the matching of the objects. Appendix F includes thirteen (13) more range images which were tested for the complete recognition approach. Training of the RBF classifier was not done again, and the same classifier as shown here was used.

Objects	classifiable geons	correct geon classifications (compared with models)	incorrect geon classifications (compared with models)
horse (first)	9	7	2
cow	8	6	2
giraffe	7	4	3
doll	25	23	2
bear	6	5	1
elephant	8	6	2
cheetah	8	4	4
hippopotamus	5	4	1
kangaroo	7	6	1
horse (view 1)	9	7	2
horse (view 2)	8	7	1
horse (view 3)	7	7	0
horse (view 4)	9	7	2
pig (view 1)	6	2	4
pig (view 2)	9	4	5
pig (view 3)	9	4	5
pig (view 4)	8	4	4

Table 4.5: Summary of classifications performed by the RBF classifier over the segmented regions of the seventeen range images tested.

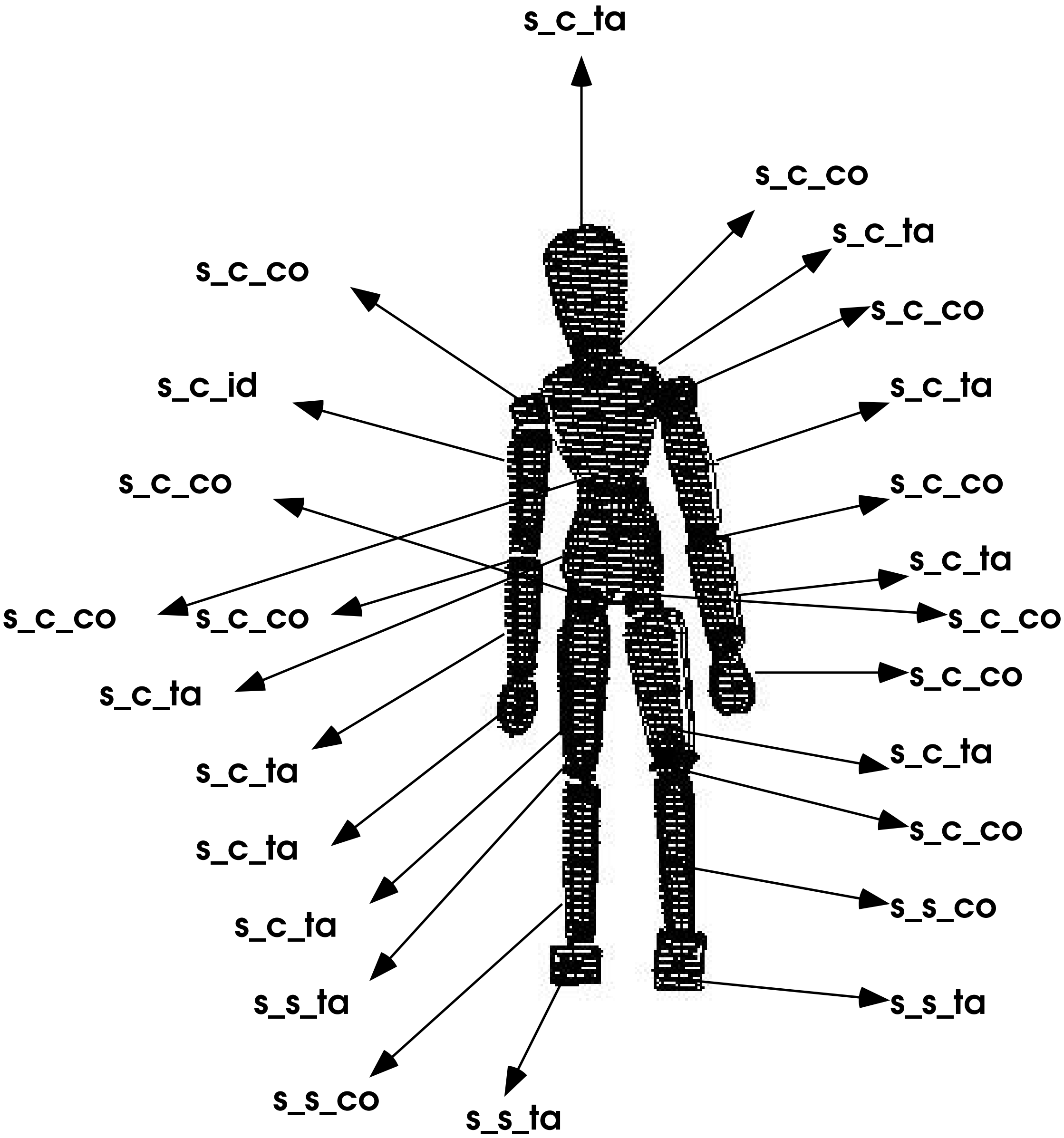


Figure 4.8: Fitted superquadric primitives shown overlayed on original 3-D data for object “doll” with geons labels from RBF classification.

Considering both the data as shown in this chapter, and the results from Appendix F, some evidence can be drawn to attest for the value of the geon classification. From the thirteen range images in Appendix F nine of them were successfully matched, these were objects “bear”, “elephant”, “cheetah”, “hippopotamus”, “kangaroo”, and “horse” in four views; and the four views of object “pig” were not matched. Noisy data and slanted views were tested and variation of the estimated superquadric parameters occurred, especially for objects “cheetah”, “horse” (all 4 views), and “pig” (all 4 views). Apart from this fact all the parts for object “horse” (all 4 views) were classified correctly by the geon-classifier, and as can be seen from Figures F.31, and F.32, it helped the matching not to be misguided. Object “cheetah” had two parts badly segmented (the “backleg” and the “tail”), and because of good classification on the other parts matching was successful. The two parts badly segmented could not be classified correctly though, because the superquadric parameters were completely misleading for those two. In the case of object “pig”, even though a successful match was not found some parts were correctly classified and even produced weak (FALSE and VERY FALSE) hypotheses for all four views.

Because small perturbations on the superquadric estimation is often expected, the geon-classification step would help insure more robustness in the recognition by accomodating these variations as weaker but sometimes feasible matches.

4.6 Chapter Summary

Raja and Jain [Raja & Jain 92] presented experiments on classifying superquadric parameters into 12 geon classes similarly to our classification problem here. They used range images of hand-carved models of geons and ran experiments with a tree classifier and k-nearest-neighbour classifiers. They presented as the best figure achieved by a tree classifier on range data from “smooth” objects an error rate of 23.3% (i.e. 76.7% correct).

From Table 4.3 the success rate achieved for the RBF classifier presented here (79.0%) is slightly better than Raja and Jain’s figure. If we count the number of times the correct classification is included in the top three probabilities of the classifier then the success rate hits 94.4%. Including more than just the highest classification is particularly interesting for our original problem of using the classification as an additional measure for the matching of three dimensional parts, because it makes it possible to include additional hypotheses with different rankings in the recognition of the objects. This issue will be explored in the next chapter where both the quantitative information of the superquadric parameters, and the qualitative classification of the geons will be used as features for designing a matching algorithm with the purpose of model-based recognition of three dimensional articulated objects.

Future work on classification of superquadrics into geons could include other fitting procedures to extract locally deformable superquadrics with more shape deformation attributes. Although this could be computationally expensive more accurate models would be acquired. Also the RBF multidimensional classifier algorithm could be used to classify different sets of volumes, not specifically geons, which could be more constrained to particular environments.

Chapter 5

Model-Based Matching through Indexing and Fuzzy Ranking of Hypotheses

This chapter is concerned with the matching approach developed in this thesis for model-based recognition. The approach is feature based, and it receives its input data from the segmentation and classification modules - Chapters 3 and 4 of this thesis respectively. Both quantitative and qualitative information about the data are explored: first by means of an indexing of the model library, based on the qualitative features; and second by a constrained search model using the quantitative features of the superellipsoid representation. The constrained search algorithm presents a novel measure of qualitative similarity, which allows ranking of the surviving hypotheses in a way suitable for evaluating class-based recognition of complex 3-D objects.

The following sections present the matching approach and discuss results and related issues in applying it to recognition using dense range images.

5.1 A Rich Volumetric Representation: Parametric Forms and Symbolic Features

A problem in 3-D model-based recognition is to find a rich and suitable representation for describing the classes of objects, which the system aims to recognize. In our opinion there is no general representation that can describe efficiently all types of shapes. Having said that, there are criteria upon which we could judge existing and newly proposed representations for 3-D recognition.

Marr [Marr 82] proposed five criteria to judge the suitability of a representation for 3-D object recognition: accessibility, scope, uniqueness, stability, and sensitivity. These criteria summarize the main requirements for a representation, however they do not provide a specific metric with which to assess the performance of a representation in a recognition task.

In general, 3-D volumetric primitives perform better than other commonly used representations such as surfaces, contours, and points in practical design issues like primitive complexity, model complexity, search complexity, reliance on verification, and model flexibility [Dickinson *et al.* 92b].

To describe a wide range of complex articulated 3-D shapes for the purpose of recognition, as we propose to do in this work, we use two volumetric representations which complement each others properties: superquadrics and geons.

Superquadrics are attractive for use in Computer Vision mainly because of their representational power, and the small number of parameters necessary to define them. However, it has been argued [Jain & Jain 89] that one of their drawbacks is the difficulty in achieving shape discrimination, or the lack of good indexing properties by their parameters. To address this problem we derive a classification of volumetric shapes (a subset of geons) and form a composite representation, superquadrics and geons. This results in a powerful representation for object recognition since we will have the representational capacity of the superquadrics and the indexing basis provided by the geons.

Chapters 3 and 4 described the segmentation and classification methods for deriving this representation. The following sections describe the matching algorithm developed in this work using this composite shape representation.

5.2 Recognition Issues and Approach

Recognition is a search problem, and as such it consists of finding the best interpretations possible for the data. Two important issues in the design of the matching process for recognition are efficiency and robustness. The first is concerned with finding the matches as quickly as possible, avoiding misleading paths sooner rather than later, and the second is concerned with error or uncertainty management where false and inexact matchings are to be dealt with.

One way to address efficiency is by splitting the search optimization problem into two related ones:

1. Minimization of the search space.
2. Minimization of the portion to be explored in the search.

In our approach we use volumetric primitives (deformable superquadrics and geons) to represent the objects and this has the effect of reducing drastically the search space in recognition, since the number of features is directly related to the size of the search [Grimson 90]. In order to minimize the fraction of the search space to be explored in the matching process, one general strategy would be to group in a meaningful way the objects in the database, so common characteristics shared by subsets of the objects would be first identified and then trigger the search in subsets of the database.

For a matching process to be able to deal with objects with similar but not identical geometric properties, and with features that may have a small range of values, a mechanism to rank or to measure the quality of the matching is more appropriate than usual binary (*i.e.* matches/fails) measures. In this case inexact matchings have to be considered, and often a categorization of the valid hypotheses is also desirable.

Figure 5.1 shows a functional diagram of the matching process developed in this thesis in order to achieve recognition. The approach is structured to favour efficiency and robustness, and works particularly well in the domain of this work.

5.3 Form of the Models

Appendix D shows a complete model as used in this work (the model for "doll"), and it contains a description of the data structures for it. All the models used in this work were created manually using the THINGWORLD [Pentland & Sclaroff 91] modelling

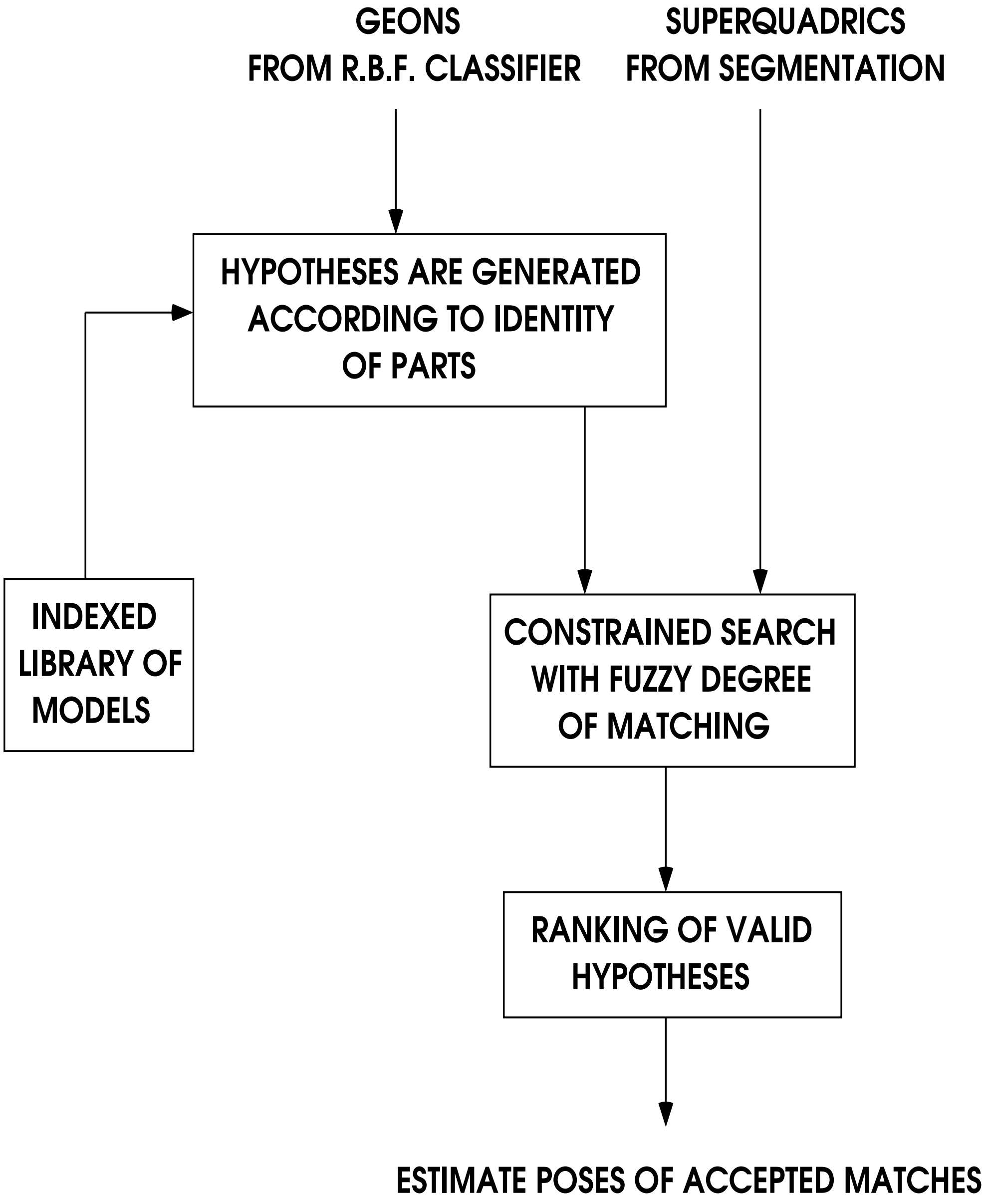


Figure 5.1: Functional diagram of the matching approach developed in this work.

system, to shape and position the superquadrics, and then their descriptions (similar to the one shown in Appendix D) were edited and placed in the database of models. Figures 6.3, 6.4, 6.5, 6.6, 6.7, 6.8, 6.9, 6.10, 6.11, 6.12, 6.13, 6.14, F.28, F.29, F.30, F.31, and F.32 of this thesis show models overlayed on the 3-D data as a result of matching and localization processes.

The geometric models are hierarchical assemblies of volumetric features. The model primitives are superquadrics, parameterized and described by:

- ϵ_1 parameter of the superquadric;
- ϵ_2 parameter of the superquadric;
- a_x, axis in x of the superquadric;
- a_y, axis in y of the superquadric;
- a_z, axis in z of the superquadric;
- t_x tapering deformation in x;
- t_y tapering deformation in y;
- *rad_bend* radius of bending deformation of the superquadric;

The model structure hierarchy is formed by placing the primitives relative to each other, relative to a global coordinate system, specifying the transformation by X, Y, and Z for translation, and R, S, and T for the rotation components. Moveable joints are specified by labelling a part allowed to move, and a distance limit (measured as an euclidean distance between the mass centres of the connected parts) which specifies the maximum relative movement. A possible extension could be to constrain joints to *e.g.* allow only motion in a plane rather than a bounded full six degrees of freedom. Additional information in the model is primitive adjacency, indicating which parts are adjacent to each other, and labels for the geon classes the primitives belong to.

5.4 Structuring the Library of Models

In model-based recognition proper organization of the models in the library is essential in order for recognition to be achieved without having to explore all possible matches. The organization of the library should in this case reflect knowledge and shared properties of the models which can be readily explored by the matching algorithm.

The approach taken in this work is to arrange the models by using the qualitative shape information of each model's parts. The indexing keys used are the identities of the parts according to one of the twelve geons outlined in Chapter 4. This structuring turns out to be an appropriate method to explore the class property in the models, and it has the capacity of retaining efficiency in a search with large libraries.

Figure 5.2 shows how the model library is organized. The indexing keys (G01, G02, G03, G04, G05, G06, G07, G08, G09, G10, G11, G12) represent the twelve geons, which are the symbolic features for the parts of the models. The models containing a specific part are linked to the indexing key for that part.

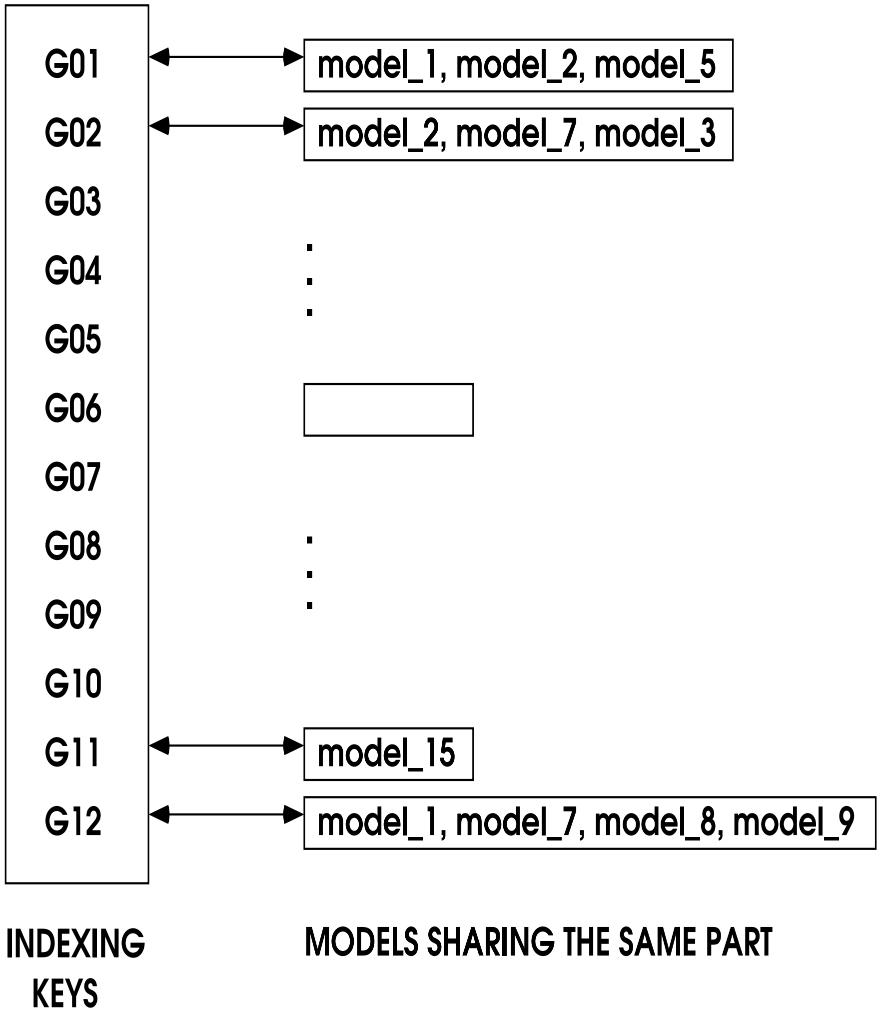


Figure 5.2: Schematic figure of the model library structure showing the geon classes as indexing keys and hypothetical models that share the pointed geons as parts.

5.5 The Need for Inexact Matching: Ranking Hypotheses Between True and False

When a hypothesis is generated, as in an object recognition problem like the one dealt with in this work, it is based on evidence extracted from the image and from the knowledge built into the system. A binary evaluation of this hypothesis, being either true or false only, overshadows the fact that because of noise, occlusion, failures or deviations in the segmentation, and geometric similarity rather than identity between the objects, a match is never exact in a real situation. A quality measure of similarity in the matching process provides a more robust and realistic evaluation of the hypotheses generated, since it can order the hypotheses within the range of 0 - 100% true. Fast information integration capabilities are also required in deriving this measure. We propose to derive such a measure from Possibility Theory, or Theory of Fuzzy Sets [Zadeh 78].

5.5.1 A Fuzzy Degree of Similarity

Possibility theory is an extension of the theory of Fuzzy Sets, and it was devised to express a multivalued logic with precision and accuracy. In contrast with a binary logic a multivalued logic permits one to deal with variations of uncertainty in a decision process.

The Formal Background

The formal background of the fuzzy reasoning methodology can be divided into three separate steps:

1. **Definition of the feature space**

   Consider a feature space $\mathcal{F}$, consisting of f coordinates such that

   $$\mathcal{F} \quad = \quad \mathcal{F}_1 \times \ldots \times \mathcal{F}_f$$

   where $\mathcal{F}_i$ is the range of allowed values for the i^{th} feature. A generic object $\mathcal{O}$ from a model library $\mathcal{ML} = \{\mathcal{O}\}$ is represented by a vector of measurements in this feature space as $[f_1, ..., f_n]$. In order to compare geometric objects as vectors of measurements we need to define distances Δ between the corresponding features of each object.

The space of interest will now be the space of distances defined as

$$\Delta\mathcal{F} \quad = \quad \Delta\mathcal{F}_1 \times ... \times \Delta\mathcal{F}_f$$

having elements

$$\vec{\delta f} \quad = \quad (\delta f_1, ..., \delta f_f)$$

where,

$$\delta f_i \quad = \quad distance(f_i, f'_i) \qquad f_i, f'_i \in \mathcal{F}_i$$

2. **Linguistic labelling: Defining the membership functions**

   Let X be a variable which receives values in a universe of discourse U, and l be a subset of U characterized by a membership function μ_l. A possibility distribution can be defined by:

   $$\pi_X(u) \quad = \quad Possibility\{X = u\} = \mu_l(u) \qquad u \in U$$

   The universe of discourse U is typically the f-dimensional space of distances, *i.e.*

   $$U = \quad \Delta\mathcal{F} \quad = \Delta\mathcal{F}_1 \times ... \times \Delta\mathcal{F}_f$$

   A linguistic variable $\mathcal{L}_f$ with term set $\{l_{f1}, ..., l_{fj}\}$ is introduced for each $\Delta\mathcal{F}_i$, where each term l_{fj} denotes a fuzzy set in $\Delta\mathcal{F}_f$ with membership function $\mu_{l_{fj}}(\delta f_f)$. The global linguistic label l will be an element of $\mathcal{L} = \mathcal{L}_1 \times ... \times \mathcal{L}_f$ with membership function as

   $$\mu_l(\delta f) \quad = \quad minimum(\mu_{l_{1j}}(\delta f_1), ..., \mu_{l_{fj}}(\delta f_f))$$

   The definition of l is seen as a fuzzy relationship $\mathcal{R}_1$ in $\Delta\mathcal{F} \times \mathcal{L}$, characterized by a membership function $\mu\mathcal{R}_1(\delta f, l)$.

3. **Evaluation of class membership**

   Similarity classes in this case are fuzzy sets in the universe of discourse of fuzzy labels. The evaluation of this similarity is characterized by a fuzzy relationship $\mathcal{R}_2$

in $\mathcal{L} \times \mathcal{C}$, where $C = \{c_1, ..., c_c\}$ is the set of similarity classes. $\mathcal{R}_2$ is characterized by a membership function $\mu\mathcal{R}_2(l, c_i)$.

The compositional rule [Zadeh 78] is the inference mechanism that gives the degree of possibility with which an object is assigned to a similarity class.

$$\mu\mathcal{R}_1 \circ \mathcal{R}_2(\delta f, c_i) \quad = \quad maximum_l \;\; minimum\{\mu\mathcal{R}_1(\delta f, l), \mu\mathcal{R}_2(l, c_i)\}$$

5.6 Finding and Ranking Correspondences: Constrained Search with Fuzzy Degree of Matching

One suitable way to organize the search is to pose the problem as a constrained search of a tree of interpretations, known as the Interpretation Tree approach.

The Interpretation Tree approach can be seen as an instance of the hypothesize-and-test paradigm in Artificial Intelligence [Nilsson 80]. At each level down of the tree, going from the root to the leaf nodes, a different feature is tested against the model, thus building up the feasible matchings between data and model. Other researchers have used the Interpretation Tree in Computer Vision, most notably [Grimson & Lozano-Perez 84, Faugeras & Hebert 86, Grimson 90, Flynn & Jain 91]. For a performance analysis of several variations of search algorithms using the Interpretation Tree approach one is referred to [Fisher 94].

5.6.1 Definitions and Notations

Let $\mathcal{D}$ be a set of all data primitives (volumetric parts) found in the scene:

$$\mathcal{D} \quad = \quad \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_d\}$$

Let $\mathcal{M}$ be a particular model being matched with subcomponents $\mathcal{M}_i$:

$$\mathcal{M} \quad = \quad \{\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_m\}$$

A valid interpretation or hypothesis ($\mathcal{I}$) of the scene will be a tuple:

$$\mathcal{I} = \quad \langle \mathcal{O}, \mathcal{B} \rangle \quad = \langle object, \{(\mathcal{D}_1, \mathcal{M}_{i1}), (\mathcal{D}_2, \mathcal{M}_{i2}), ..., (\mathcal{D}_j, \mathcal{M}_{ij})\}\rangle$$

where $\mathcal{O}$ is the identity of the object in the model library and $\mathcal{B}$ is an ordered set of the associations found between the data primitives and the model primitives. A verification stage including the pose estimation of the model is done after valid interpretations of the data are found and ranked. A variation of the Interpretation Tree approach estimates the transformation (rotation and translation) between data and model while building the interpretation [Grimson & Lozano-Perez 84, Faugeras & Hebert 86]. This is a powerful constraint, in particular when using features like edges and surfaces which generate thousands of possible bindings to be searched. In our case, because we are using volumetric primitives, and we have introduced an indexing scheme to organize the models and narrow down the initial hypotheses, we deal with a small number of possible interpretations (tens or hundreds). The surviving hypotheses are therefore ranked and passed to the pose estimation module which introduces a new method to calculate poses of articulated objects represented by volumetric primitives. This approach presented here is efficient and it is specially suited for our problem of recognizing similar complex objects.

5.6.2 Geometric Constraints and Design Features

The efficacy of the Interpretation Tree approach as a recognition algorithm is based on the use of geometric constraints to prune the branches of the tree which lead to invalid hypotheses. These constraints are forced by unary and binary predicates, which are applied sequentially to every hypothesis generated as the matching grows by adding new bindings. If a predicate is not satisfied the interpretation is discarded, and the subtree is pruned.

Unlike the usual Interpretation Tree, which uses a bivalued predicate for evaluating the geometric constraints in the search, we develop a Fuzzy Predicate with five (5) fuzzy sets indicating possible evaluations of the evidence. This allows us to deal with uncertainty in the matching in a more robust way, and also to have a ranking of surviving hypotheses, which proves to be a distinctive measure in matching objects with similar geometric properties.

In the remainder of this subsection we describe unary and binary fuzzy predicates, and the special design features of our Interpretation Tree algorithm respectively. The details of the fuzzy predicate evaluation, together with the quality measure of similarity are given in Section 5.5.3.

Fuzzy Unary Predicates

In our approach we use four different fuzzy unary predicates:

- **The Volume Predicate** $\mathcal{FP}_{vol}(\mathcal{I})$: Let $V_{\mathcal{D}_i}$ be the estimated volume of the part $\mathcal{D}_i$, and let $V_{\mathcal{M}_{ni}}$ be the maximum volume prescribed for $\mathcal{M}_{ni}$. The volume predicate is:

$$\mathcal{FP}_{vol}(\mathcal{I}) \quad = \quad \mu_\delta(\mathcal{FP}_{vol}) \tag{5.1}$$

- **The Geon Type Predicate** $\mathcal{FP}_{geon\_type}(\mathcal{I})$ is evaluated calling

$$\mathcal{FP}_{geon\_type}(\mathcal{I}) \quad = \quad \mu_\delta(\mathcal{FP}_{geon\_type}) \tag{5.2}$$

  and checks the similarity between the types.

- **The Shape Intrinsic Parameter Predicate** $\mathcal{FP}_{shape\_intrinsic}(\mathcal{I})$: This predicate is applied after the $\mathcal{FP}_{geon\_type}$ predicate to evaluate the five shape parameters of the deformable superquadric that identify the geon $(\epsilon_1, \epsilon_2, t_x, t_y, rad\_bend)$.

- **The Part Scale Predicate** $\mathcal{FP}_{part\_scale}(\mathcal{I})$: This predicate evaluates the ratio between the longest axis, and the shortest axis (*i.e.* from the a_1, a_2, a_3) of the two volumetric primitives.

Fuzzy Binary Predicates

We use three binary predicates:

- **The Adjacency Predicate** $\mathcal{FP}_{adjacency}(\mathcal{I})$: This predicate returns the evaluation of $\mu_\delta(\mathcal{FP}_{adjacency})$ for each pair of associations about their adjacency in the data and in the model.

- **The Distance Predicate** $\mathcal{FP}_{distance}(\mathcal{I})$: This predicate checks the Euclidean distance between each pair of associations made to see if they agree within a specified tolerance; a membership degree evaluation of this distance is then calculated by $\mu_\delta(\mathcal{FP}_{distance})$. The distance is measured between the mass centres of each volumetric part. In the case where the part is articulated the tolerance given is higher in order to comply with the allowed range of distance between the two parts.

- **The Parts Proportion Predicate** $\mathcal{FP}_{proportion}(\mathcal{I})$: This predicate evaluates the proportion between the biggest and smallest parts of the model and data to provide discriminative information in the case of similar objects.

Design Features

- **NULL Associations**: In order to allow for spurious data, perhaps from bad segmentation or fitting, we provide the Interpretation Tree with NULL associations [Grimson 90], which gives the capacity, during the search, to ignore any data primitive $\mathcal{D}_i$ not found in the model.
- **Data Ordering**: The data primitives $\mathcal{D}_i$ are ordered starting from the biggest parts until the smallest before being passed to be searched. This helps to improve the efficiency of the search.

5.6.3 Evaluating Similarity in a Match by a Fuzzy Measure

Feature values extracted from image data, such as parameters of the deformable superquadrics fitted in the segmentation procedure from Chapter 3, are evidence for a matching procedure and as such there are variations in the degree of uncertainty for these features. When evaluating a geometric constraint for (data primitive, model primitive) binding, usually a tolerance is prescribed for each predicate test and the matching result indicates only the acceptance or not (*i.e.* true or false) of that binding. In this thesis we aim to recognize complex 3-D objects that are similar in their geometric properties and are organized into classes. This brings to the match evaluation the task of measuring different degrees of similarity, hence the need for a quality measure of similarity.

Figure 5.3 shows a diagram displaying the stages where the predicates and similarity evaluation are applied in the matching algorithm. The example shown in Figure 5.3 is of hypothetical model and data primitives; results of the application of this algorithm are shown in further sections.

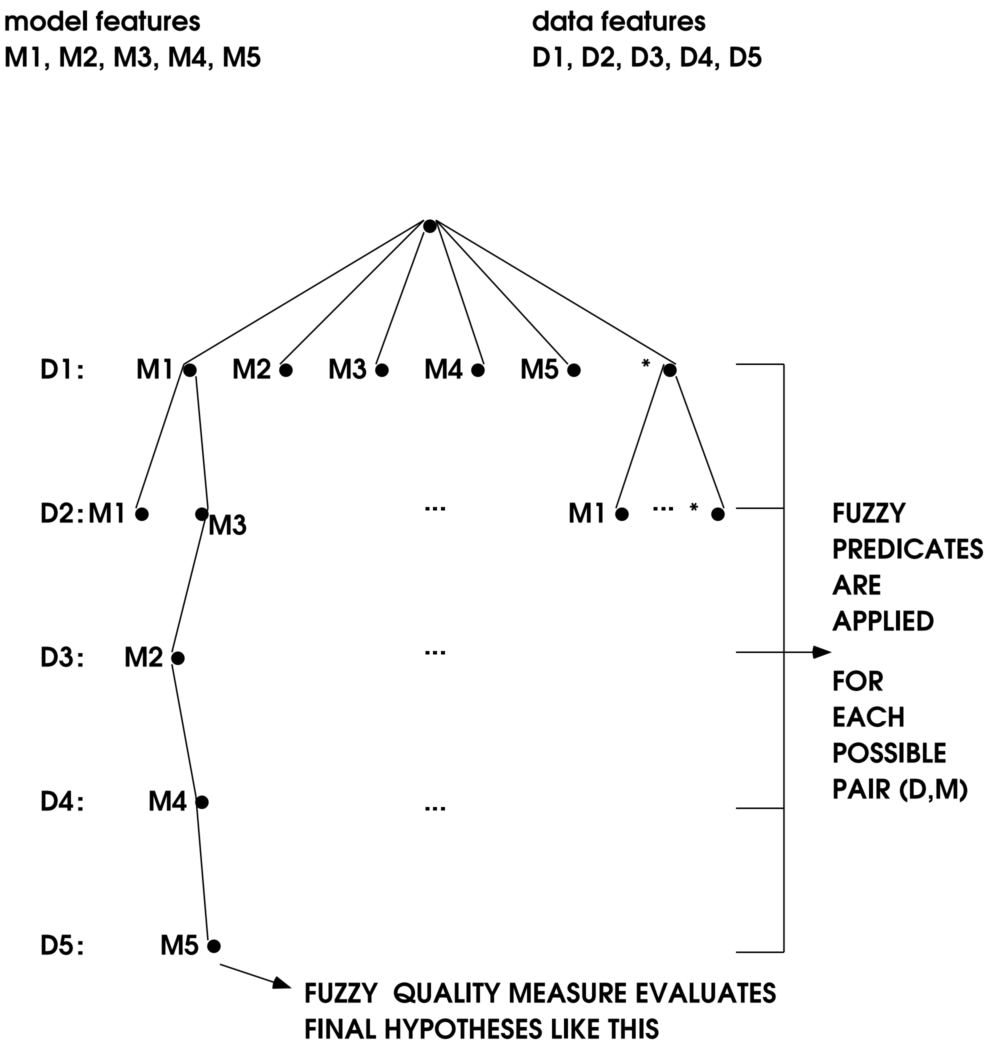


Figure 5.3: Interpretation tree search model showing where the evaluation of the fuzzy predicates and the fuzzy degree of similarity apply.

The matching algorithm, including the evaluations of the fuzzy predicates and the final degree of similarity, is divided into the following steps:

1. Compute normalized distances between the pair of corresponding features while analyzing the current binding (M_i, D_j), using:

$$\delta(fm_q fd_q) = \|1.0 - (fm_q - fd_q)/fm_q\| \quad 1 \leq q \leq n \tag{5.3}$$

where n is the number of features involved amongst all predicates, in this case $n = 11$ (7 predicates, but the third unary predicate provides actually 5 feature evaluations).

2. Evaluate the degree of membership of each distance feature $\Delta\mathcal{F}$ producing fuzzy inputs. The membership functions are defined in Figure 5.4. For each distance feature computed, a degree of membership μ (between 0.0 and 1.0) and a linguistic label (VERY FALSE, FALSE, ACCEPTABLY TRUE, TRUE, VERY TRUE) are associated to that feature matching. This can be written as the following fuzzy relationship $R_1 : \Delta\mathcal{F} \times \mathcal{L}$,

$$\mu R_1([\delta f_1, \delta f_2, ..., \delta f_{11}], [l_1, l_2, ..., l_5]) = minimum_i[\mu_{lf}(\delta f_i)] \qquad (5.4)$$

where $[\delta f_1, \delta f_2, ..., \delta f_{11}] \in \Delta\mathcal{F}$, and $[l_1, l_2, ..., l_5] \in \mathcal{L}$.

3. The actual fuzzy predicates are thresholded to indicate acceptable matchings (the ones that satisfy the constraints). The acceptable values are ACCEPTABLY TRUE, TRUE, and VERY TRUE.

4. Accepted pairings are given a fuzzy degree of similarity, which is computed as follows. Weights are given according to the values shown in Tables 5.1 and 5.2 in order to express the relevance of each feature matched and its degree of membership. The values were determined empirically, reflecting our experience with the tests on the importance of each feature. A membership function $R_2 : \mathcal{L} \times Sim\_set$ is defined as:

$$\mu R_2[l_{f1}, l_{f2}, ..., l_{f11}, Sim\_set] = \frac{[\sum_{i=1}^{11} wf[i] \times wt[j]]}{maximum_j(\sum_{i=1}^{11} wf[i] \times wt[j])} \qquad (5.5)$$

where, $j \in [1, 2, 3, 4, 5]$ (*i.e.* 5 fuzzy sets), and Sim_set is chosen to be the same set as in Figure 5.5.

5. The quality measure of similarity is computed by combining the fuzzy relationships R_1 and R_2 by a composition rule of Fuzzy Sets [Zadeh 78]:

$$\mu R_1 \circ R_2([\delta f_1, \delta f_2, ..., \delta f_{11}], Sim\_set) =$$
$$maximum_l \;\; minimum[\mu R_1([\delta f], [l]), \mu R_2(l_f, Sim\_set)] \qquad (5.6)$$

6. Steps 1 to 5 are repeated until all the data parts are evaluated. The surviving hypotheses are then ranked as follows. The tree evaluation produces np fuzzy quality measures, one for each part, and a global similarity measure is computed by evaluating a membership function $R_3 : \mathcal{L} \times Sim\_set$, defined as:

$$\mu R_3[l_{part_1}, l_{part_2}, ..., l_{part_{np}}, Sim\_set] = \frac{[\sum_{i=1}^{np} wpart[i] \times wt[j]]}{maximum_j(\sum_{i=1}^{np} wpart[i] \times wt[j])} \quad (5.7)$$

with Sim_set as in Figure 5.5, and the weights for the parts $wpart[i]$ are set as equal to 1 for all the np parts.

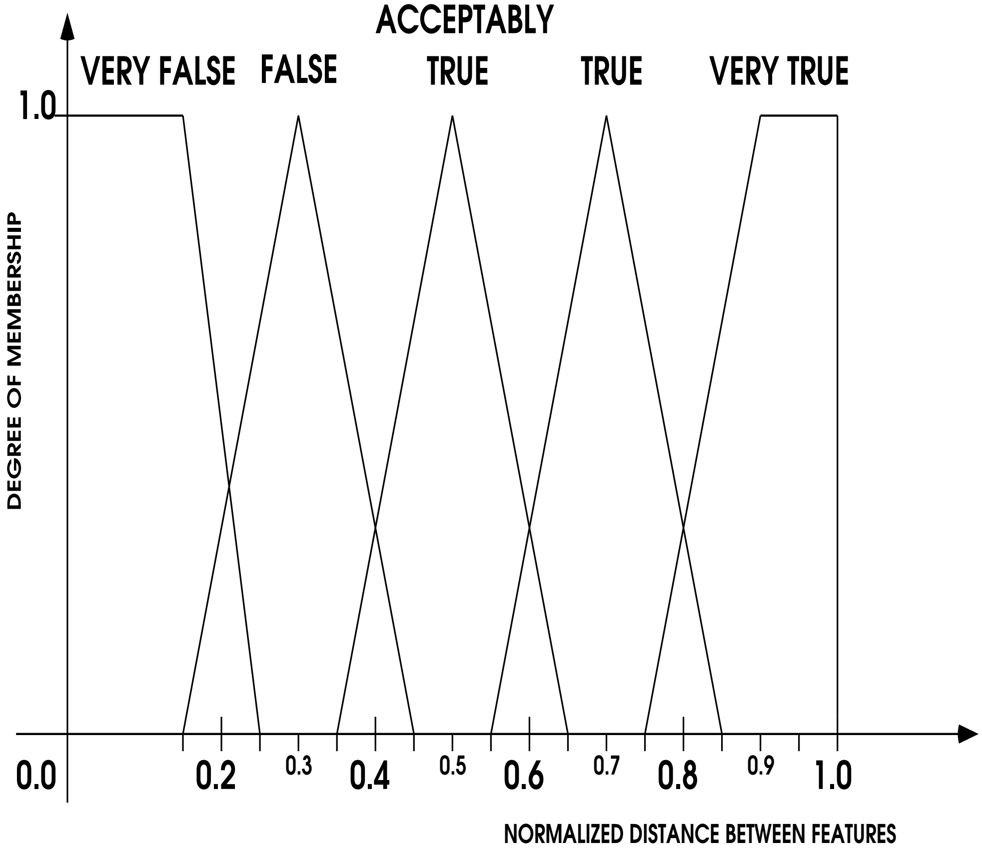


Figure 5.4: Membership functions of the fuzzy sets proposed to evaluate the degree of similarity in the matching.

5.7 Experiments and Results

This section presents results on matching complex 3-D objects from range data against a model library of 16 objects. A list of the models is given in Table 5.3, an example

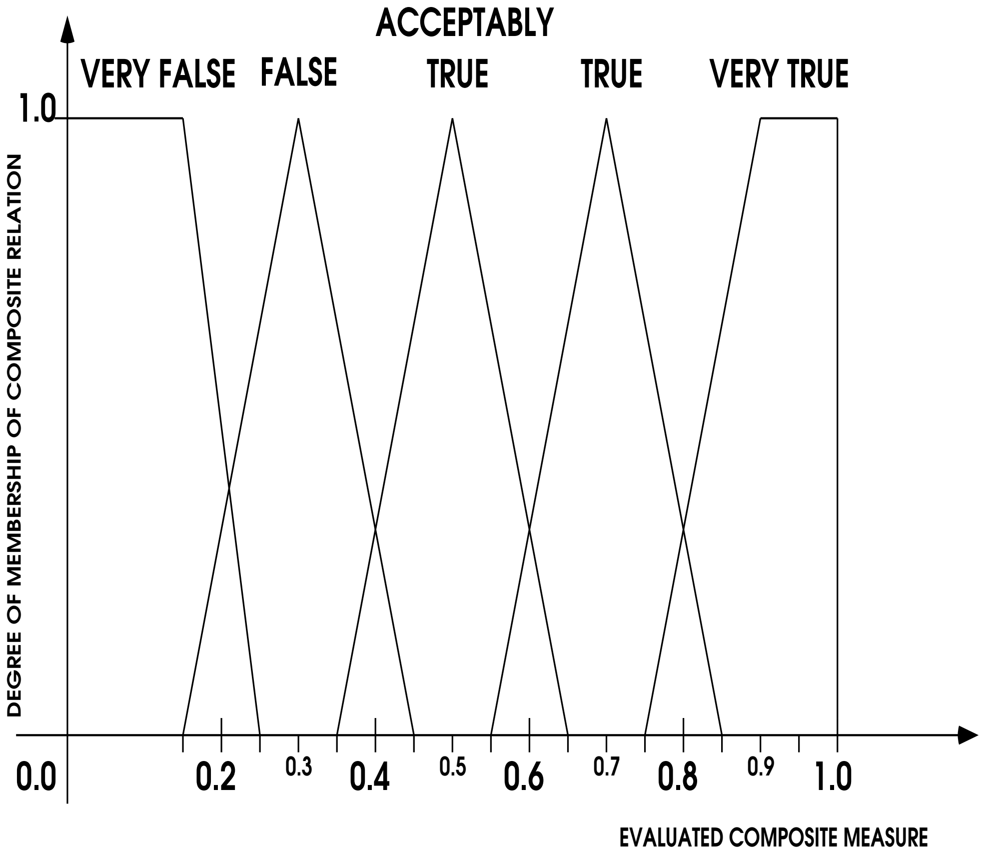


Figure 5.5: Similarity sets defined to provide equivalent ranges (linguistic terms) for fuzzy degree of matching.

wf_1	wf_2	wf_3	wf_4	wf_5	wf_6	wf_7	wf_8	wf_9	wf_{10}	wf_{11}
2.0	1.0	1.0	1.0	1.0	1.0	1.0	2.0	3.0	3.0	3.0

Table 5.1: Weights expressing the relevance of each feature (wf_i) in the similarity evaluation.

from the model library is given in Appendix D. The model library is indexed by the qualitative volumetric parts of the objects, as in Figure 5.2.

wt_1	wt_2	wt_3	wt_4	wt_5
2.0	1.5	1.0	0.5	0.2

Table 5.2: Weights expressing the relevance of each fuzzy set (linguistic term) (wt_j) in the similarity evaluation.

number	MODEL
1	mod1_doll
2	mod2_cow
3	mod3_pig
4	mod4_cheetah
5	mod5_duck
6	mod6_nessie
7	mod7_horse
8	mod8_lion
9	mod9_bigbird
10	mod10_giraffe
11	mod11_whale
12	mod12_elephant
13	mod13_hippo
14	mod14_bear
15	mod15_pelican
16	mod16_kangaroo

Table 5.3: List of models used in these experiments.

Range images acquired were processed by the segmentation and classification procedures, described in Chapter 3 and Chapter 4 respectively. The direct input to the matching procedure is an array of measurements with the quantitative features of the superquadrics (15 features), and the qualitative feature identifying the geon category for each volumetric part successfully segmented in the range image.

Results are shown for four different objects as described in previous chapters: a wooden doll, and plastic miniatures of a horse, a cow, and a giraffe.

Figure 5.6 shows a segmented image of the object "horse" with labels indicating the data primitives in the order used by the Interpretation Tree algorithm. The data primitives are those extracted from the superquadric segmentation as given in Chapter 3 and as classified into geons as given in Chapter 4. Table 5.4 gives the classification and the matching results for object "horse".

Figures 5.7, 5.8, 5.9 show objects "cow", "giraffe", and "doll", respectively, with labels indicating the data primitives in the order used by the Interpretation Tree algorithm. Tables 5.5, 5.6, 5.7 give the classification and the matching results for objects "cow", "giraffe", and "doll", in this order.

Results on Tables 5.4, 5.5, 5.6, and 5.7 show that all objects were successfully recognized and ranked differently according to the fuzzy measure of similarity. Most similar objects

Parts	**Geon Classification**
D1	s_c_co
D2	b_c_id
D3	s_c_id
D4	s_s_ta
D5	b_c_ta
D6	b_c_co
D7	**s_c_ta**
D8	s_s_ta
D9	**b_c_id**

(a)

Model Matched	**Fuzzy Measure of Similarity**	**Linguistic Label**
mod7_horse	0.82	TRUE
mod2_cow	0.61	TRUE
mod14_bear	0.54	ACCEPTABLY TRUE

(b)

Table 5.4: (a) Ordered list of data primitives with geon classifications for object "horse" (Incorrect geon classifications are shown in bold type); (b) Results showing the valid hypotheses (only the matches with rank equal to or above "ACCEPTABLY TRUE") and their final fuzzy measure of similarity.

Parts	**Geon Classification**
D1	s_c_id
D2	b_c_ta
D3	s_c_id
D4	s_c_co
D5	s_s_ta
D6	**b_s_co**
D7	s_s_co
D8	**b_s_ta**

(a)

Model Matched	**Fuzzy Measure of Similarity**	**Linguistic Label**
mod2_cow	0.92	TRUE
mod7_horse	0.61	TRUE
mod14_bear	0.48	ACCEPTABLY TRUE
mod13_hypo	0.41	ACCEPTABLY TRUE

(b)

Table 5.5: (a) Ordered list of data primitives with geon classifications for object "cow" (Incorrect geon classifications are shown in bold type); (b) Results showing the valid hypotheses (only the matches with rank equal to or above "ACCEPTABLY TRUE") and their final fuzzy measure of similarity.

Parts	Geon Classification
D1	**b_c_ta**
D2	s_c_co
D3	b_c_id
D4	s_s_co
D5	**b_s_co**
D6	b_s_ta
D7	**s_s_co**

(a)

Model Matched	Fuzzy Measure of Similarity	Linguistic Label
mod10_giraffe	0.94	VERY TRUE
mod6_nessie	0.41	ACCEPTABLY TRUE

(b)

Table 5.6: (a) Ordered list of data primitives with geon classifications for object "giraffe"; (b) Results showing the valid hypotheses (only the matches with rank equal to or above "ACCEPTABLY TRUE") and their final fuzzy measure of similarity.

Parts	Geon Classification
D1	s_c_ta
D2	s_c_ta
D3	s_c_ta
D4	s_c_ta
D5	s_c_ta
D6	s_s_co
D7	s_s_co
D8	s_c_ta
D9	s_c_ta
D10	s_c_ta
D11	**s_c_id**
D12	s_s_ta
D13	s_c_co
D14	**s_c_co**
D15	s_c_ta
D16	s_s_ta
D17	s_c_co
D18	s_c_co
D19	s_c_co
D20	s_c_co
D21	s_s_ta
D22	s_c_co
D23	s_c_co
D24	s_c_co
D25	s_c_co

(a)

Model Matched	Fuzzy Measure of Similarity	Linguistic Label
mod1_doll	0.89	VERY TRUE

(b)

Table 5.7: (a) Ordered list of data primitives with geon classifications for object "doll" (Incorrect geon classifications are shown in bold type); (b) Results showing the valid hypotheses (only the matches with rank equal to or above "ACCEPTABLY TRUE") and their final fuzzy measure of similarity.

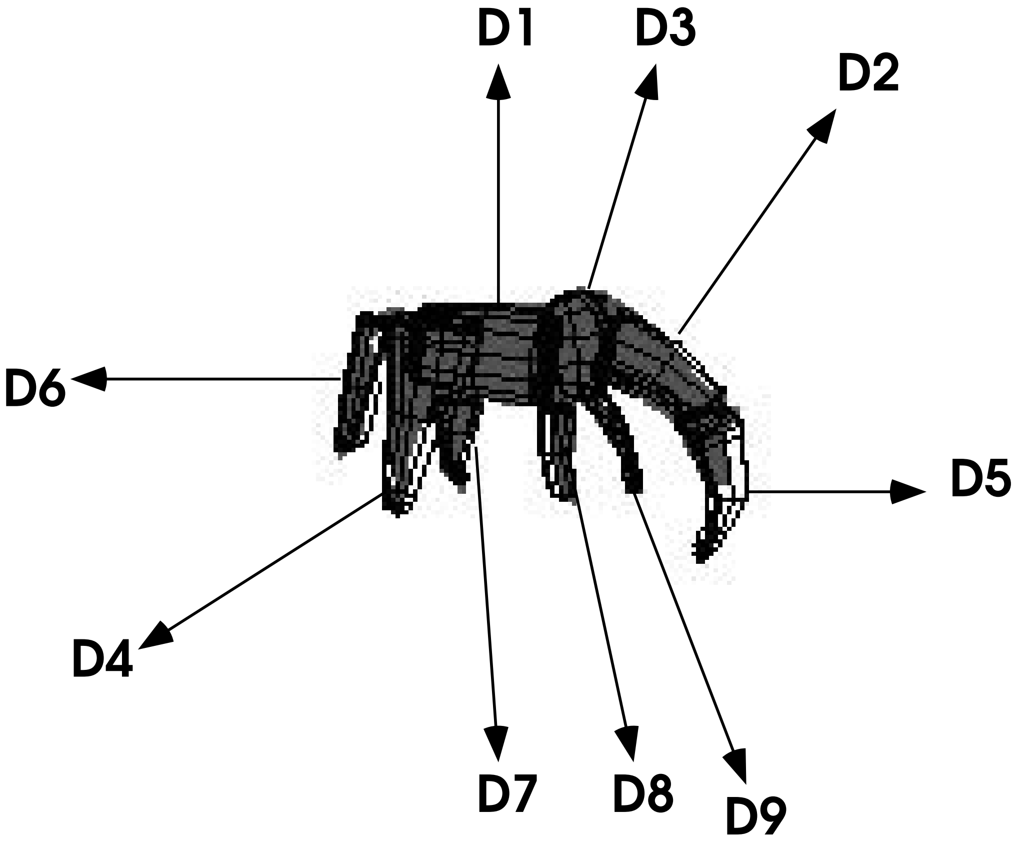


Figure 5.6: Segmented image of the object "horse" with the data features (part primitives) labelled in the order they were passed to the constrained search algorithm.

tested, the "horse" and the "cow", indicate good discriminative performance of the method by not only ranking them as the right object, but also presenting an order of other valid hypotheses above the threshold of (AT= Acceptably True).

Figures 5.10 and 5.11 show complete successful paths, and branching, of the Interpretation Tree for the objects "horse" and "doll", respectively. These figures show only the branches that are accepted by the fuzzy predicates, although for each data primitive being tested (*e.g.* D_d) all the model primitives in the list (*e.g.* M_m) are paired. The numbers shown next to the branches indicate which model primitives had passed the fuzzy predicates with at least value "AT= Acceptably True". Figures 5.10 and 5.11 show only the main path explored for the "horse" model, and the "doll" model respectively.

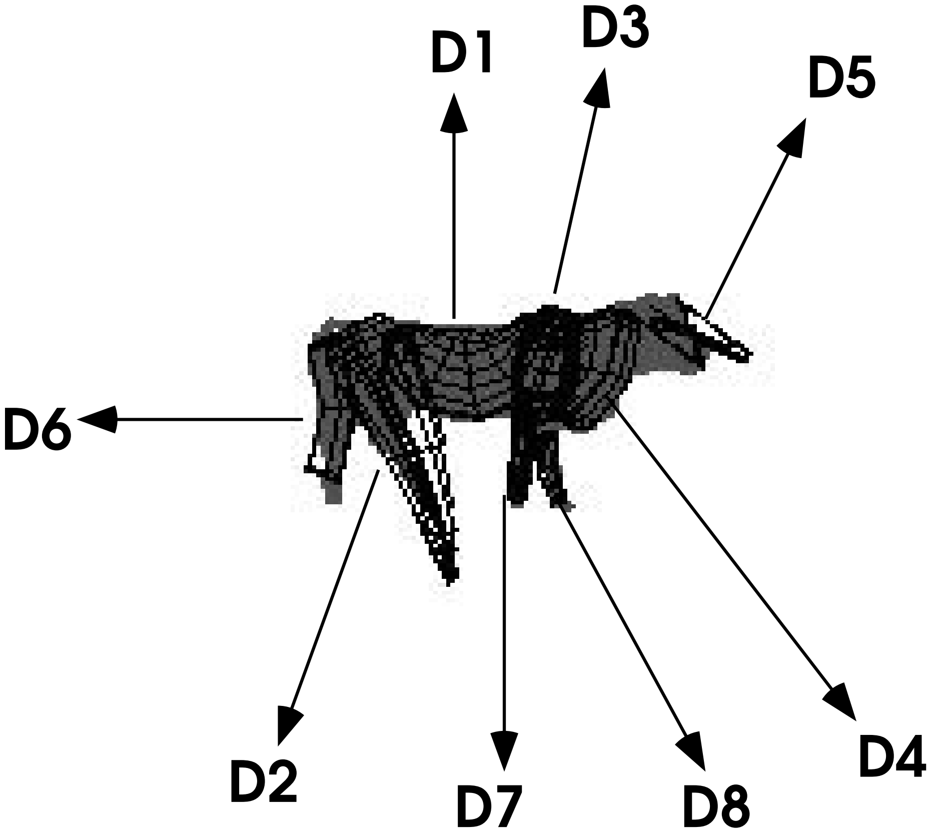


Figure 5.7: Segmented image of the object "cow" with the data features (part primitives) labelled in the order they were passed to the constrained search algorithm.

The data primitives ($\mathbf{D}_n$) are ordered according to size, from the largest part to the smallest, and the model primitives are ordered in the sequence the parts are described in the model. The preliminary successful pairings at each level of the tree are shown, and pruning is indicated by no further expansion of the nodes. Preliminary pairings are chosen by applying the Geon Type Predicate, and pruning occurs with the evaluation of all the unary and binary fuzzy predicates. Appendix D shows the complete model for object "doll". Table 5.8 shows the reduction in the tree size for all the four objects achieved by the pruning constraints and predicates. The number listed in the last column of Table 5.8 corresponds to the nodes searched for the correct model only.

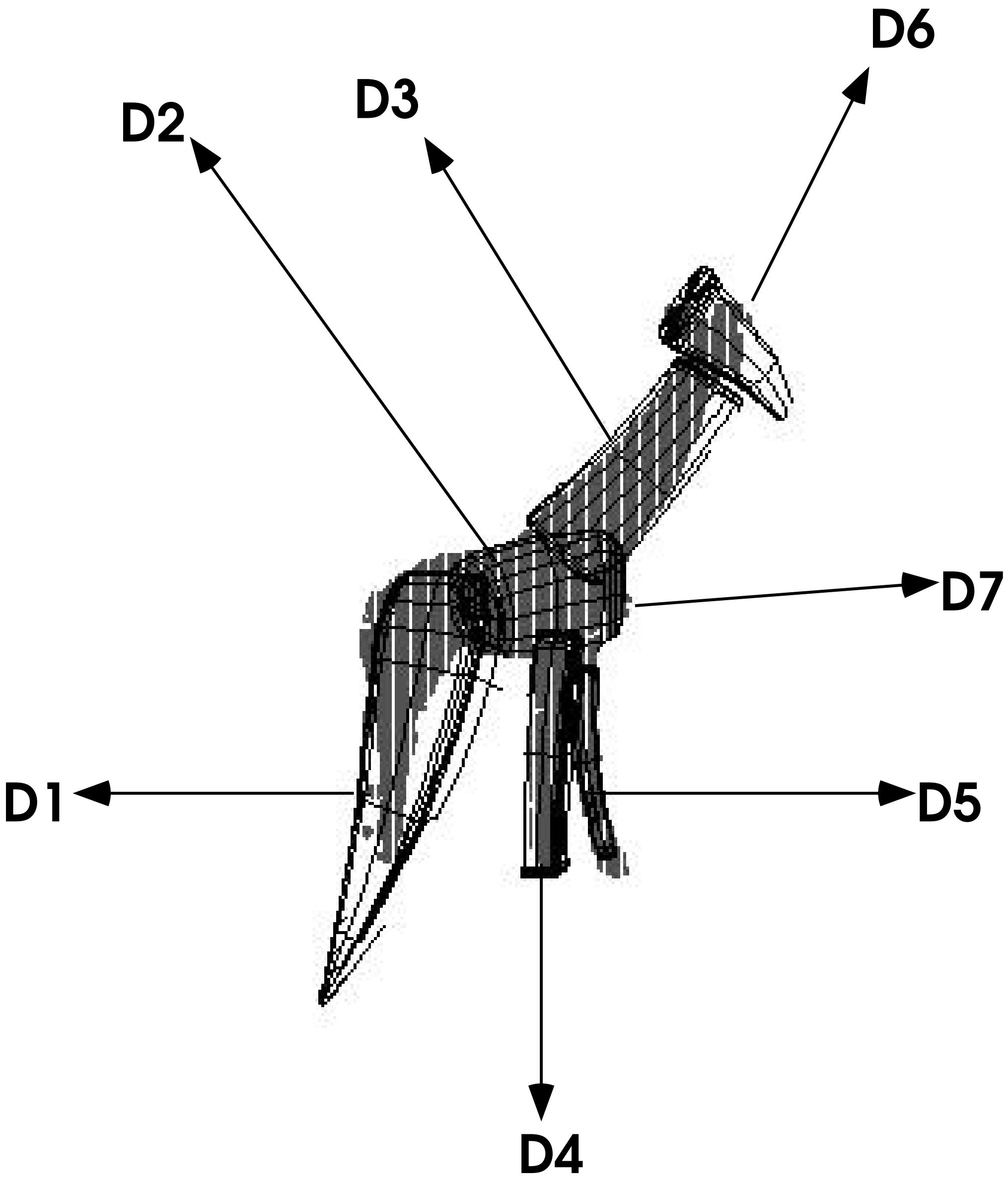


Figure 5.8: Segmented image of the object "giraffe" with the data features (part primitives) labelled in the order they were passed to the constrained search algorithm.

5.8 Discussion and More Experiments

In addition to the test objects and scenes shown in this chapter, Appendix F contains thirteen (13) more range images of seven (7) different objects, and their respective output results on the stages of recognition.

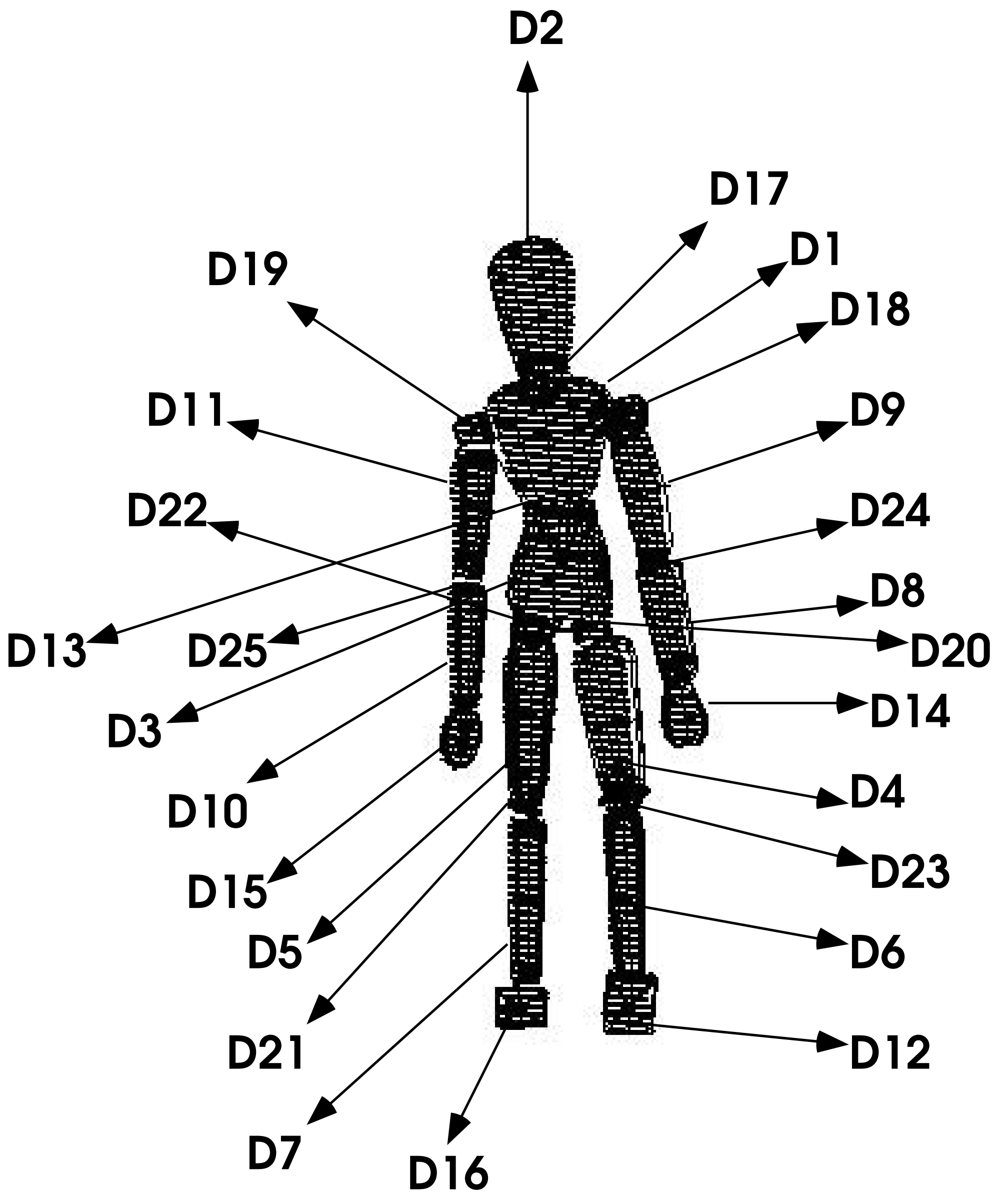


Figure 5.9: Segmented image of the object "doll" with the data features (part primitives) labelled in the order they were passed to the constrained search algorithm.

Objects "bear", "elephant", "cheetah", "hippopotamus", and "kangaroo" were tested in a single viewing position, and objects "horse" and "pig" in four different views for each.

Matching was successful for objects "bear", "elephant", "cheetah", "hippopotamus", "kangaroo" , and "horse"; and it did not find the correct hypothesis for the object

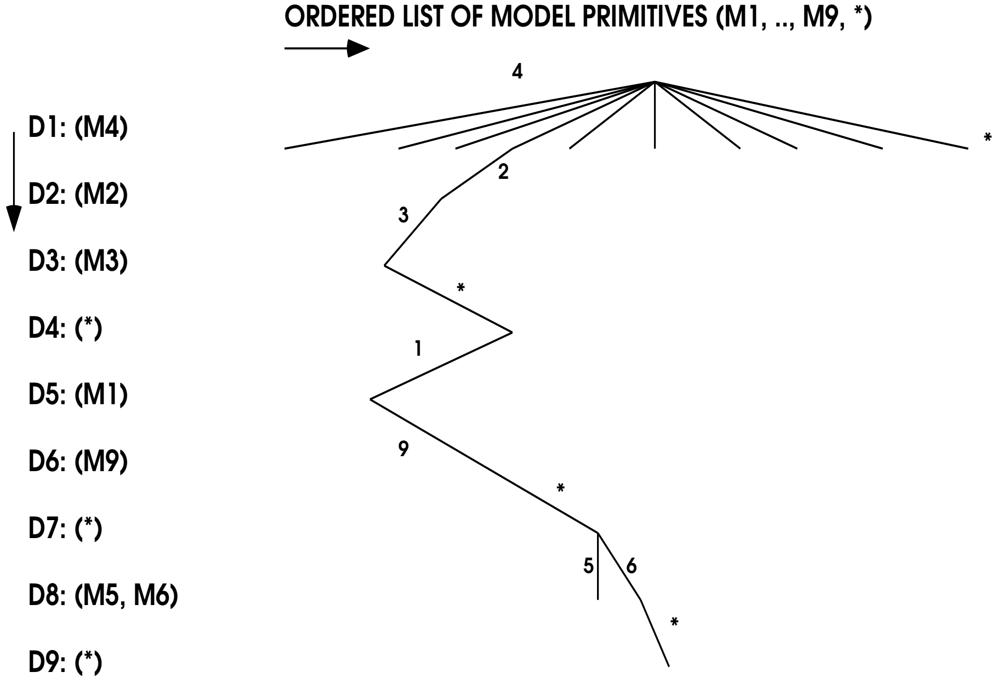


Figure 5.10: Portion of the Interpretation Tree for object "horse" showing the successful path and branches explored at each level.

object	**number of nodes for full tree**	**number of nodes actually explored**
doll	2.46×10^{35}	1274
horse	1.11×10^{9}	90
cow	48,427,561	96
giraffe	2,396,745	98

Table 5.8: Number of possible nodes for the full tree, and number of nodes actually explored for each object tested.

"pig" for all the views taken. Table 5.9 shows the ordered data primitives, Table 5.10 shows the valid hypotheses with the respective fuzzy degrees of similarity for the nine (9) images with successful matches, and Table 5.11 gives the highest measures achieved for the object "pig". Table 5.12 gives the number of nodes explored for the correct model, for each image.

Only in the cases of the unsuccessful matches of the object "pig" (Figures F.33, F.34) does the searching appear to have been exploring the nodes at random. For the other objects and test scenes the number of nodes explored for the correct models was small,

Object	**Part D1**	**Part D2**	**Part D3**	**Part D4**	**Part D5**	**Part D6**	**Part D7**	**Part D8**	**Part D9**
bear	s_c_co	s_s_co	s_c_ta	s_s_co	s_s_co	**s_s_co**	–	–	–
elephant	s_c_co	b_c_ta	s_c_co	s_s_co	s_s_co	**b_c_ta**	s_c_co	**s_s_co**	–
cheetah	b_c_id	s_s_co	s_s_co	**s_s_id**	**s_c_ta**	**s_c_ta**	s_s_co	**s_s_co**	–
hippopotamus	s_c_co	s_s_co	**s_c_ta**	s_s_co	s_s_co	–	–	–	–
kangaroo	s_c_co	b_c_id	b_c_id	b_s_ta	s_s_co	b_c_co	**s_c_ta**	–	–
horse (view 1)	s_c_co	b_c_id	s_c_id	s_s_ta	b_c_ta	b_c_co	**s_c_ta**	s_s_ta	**b_c_id**
horse (view 2)	s_c_co	b_c_id	s_c_id	s_s_ta	b_c_ta	b_c_co	**s_c_ta**	s_s_ta	–
horse (view 3)	s_c_co	b_c_id	s_c_id	s_s_ta	b_c_ta	b_c_co	s_c_ta	–	–
horse (view 4)	s_c_co	b_c_id	**b_c_id**	s_c_id	s_s_ta	b_c_ta	b_c_co	**s_c_ta**	s_s_ta
pig (view 1)	s_c_co	s_c_ta	**s_c_ta**	**s_c_ta**	**s_c_co**	**b_c_co**	–	–	–
pig (view 2)	s_c_co	s_c_co	s_c_id	**s_c_ta**	**s_c_ta**	b_c_co	**s_c_ta**	**s_c_ta**	**s_c_co**
pig (view 3)	s_c_co	s_c_co	s_c_id	**s_c_ta**	**b_c_co**	**b_c_co**	**b_c_ta**	s_s_ta	**b_c_id**
pig (view 4)	s_c_co	b_c_id	s_c_id	**s_s_ta**	**b_c_ta**	**b_c_co**	s_c_ta	**s_s_ta**	–

Table 5.9: Ordered list of data primitives with geon classifications for all the objects in Appendix F. No relevant geons are denoted by "–". Incorrect geon classifications are shown in bold type.

and the searching was more focussed.

The top match found for objects "bear", "elephant", "cheetah", "hippopotamus", "kangaroo", and "horse" was the correct one; and in the cases where other hypotheses were also accepted with lower fuzzy degree of similarity (objects "bear", "hippopotamus", and "horse"), it was due to the actual similarity of the objects' shapes.

The successful match of object "cheetah" is an evidence of robustness of the matching, because the correct hypothesis was found even considering that two parts (the "backleg" and the "tail") were not correctly indicated from the segmentation.

No correct hypothesis for the object "pig" was found in any of the four views taken. Figures F.33, F.34 show only some of the parts that were accepted as similar, and they were drawn as an exercise for the pose estimation of individual parts. The main explanations for this are the lack of precision on the estimated superquadric parameters, and the actual difficulty in representing such an object with volumetric primitives considering the similarities of its parts.

Object	**Model Matched**	**Fuzzy Measure of Similarity**	**Linguistic Label**
bear	mod14_bear	0.62	TRUE
	mod13_hippo	0.54	A. TRUE
elephant	mod12_elephant	0.60	A. TRUE
cheetah	mod4_cheetah	0.53	A. TRUE
hippopotamus	mod13_hippo	0.51	A. TRUE
	mod14_bear	0.45	A. TRUE
kangaroo	mod16_kangaroo	0.74	TRUE
horse (view 1)	mod7_horse	0.58	A. TRUE
	mod2_cow	0.50	A. TRUE
	mod14_bear	0.45	A. TRUE
horse (view 2)	mod7_horse	0.70	TRUE
	mod2_cow	0.62	A. TRUE
	mod14_bear	0.58	A. TRUE
horse (view 3)	mod7_horse	0.68	TRUE
	mod2_cow	0.61	A. TRUE
	mod14_bear	0.52	A. TRUE
horse (view 4)	mod7_horse	0.50	A. TRUE
	mod2_cow	0.42	A. TRUE
	mod14_bear	0.40	A. TRUE

Table 5.10: Results showing the fuzzy degrees of similarity for the best hypotheses for objects shown in Appendix F. For objects “bear”, “elephant”, “cheetah”, “hippopotamus”, “kangaroo”, and “horse” (all views) all hypotheses higher than ACCEPTABLY TRUE (A. TRUE) are shown.

Object	**Model Picked**	**Fuzzy Measure of Similarity**	**Linguistic Label**
pig (view 1)	mod3_pig	0.48	FALSE
pig (view 2)	mod3_pig	0.46	VERY FALSE
pig (view 3)	mod3_pig	0.46	VERY FALSE
pig (view 4)	mod3_pig	0.46	VERY FALSE

Table 5.11: Results showing the highest measure of similarity achieved for object "pig" in all views, since the match was unsuccessful.

object	**number of nodes actually explored**
bear	91
elephant	144
cheetah	576
hippopotamus	84
kangaroo	336
horse (views 1;2;3; and 4)	260; 180; 216; 400
pig (views 1;2;3; and 4)	12,527; 50,654; 36,349; 43,002

Table 5.12: Number of nodes actually explored for the top hypothesis, for each object tested in Appendix F.

5.9 Related Work

There has not been much work in recognition of 3-D articulated objects using part primitives. Previous works have addressed mainly description and symbolic identification in simple cases. The matching technique presented in this chapter has many different features addressing issues of part-based representation, inexact matching, library indexing, and constraint based search. Here we discuss previous works that are most related to the above mentioned issues.

[Nevatia & Binford 77] described a method for recognition of 3-D curved objects. Results are shown for different objects including toys such as a "horse", and a "doll". Description of the objects is based on finding elongated axes of the object parts using the 2-D outermost boundaries of the object, and projections in eight directions of the object boundaries. The projections are examined to find cross-section lengths, and end-points of the cylinders. Recognition performs symbolic identification only, and it is divided into two stages: first, a simple indexing scheme, where the indices have to match exactly the indices from the description process, is used to make the hypothesis; second, the description from data and from the model are compared in a graph matching algorithm.

[Marr & Nishihara 78] proposed a general method for shape recognition, in which objects are represented by a set of generalized cylinders. This set of generalized cylinders is organized as a hierarchy, where smaller parts are at the lowest levels. A process to describe animal shapes in terms of generalized cylinders was presented, in which elongated axes of the objects' salient parts were found from the 2-D projections of the outline of the shape. These axes were hypothesized as the axes of cylinders describing the parts. An identification process is outlined, along with a scheme to represent relative orientation of the parts. Actual mechanisms for how to derive such a hierarchical description from images, and to achieve identification and location of objects represented in a database were not described.

[Jain & Hoffman 88] described an evidence-based recognition technique for 3-D objects. Surface patches are extracted from range images of objects, and a set of unary and binary properties are identified in the data. Evidence conditions with corresponding evidence weights for various objects are organized in a rulebase, and interpretation of the data is achieved by searching the rulebase to compare properties. The high cost in this approach was that all the objects present in the database were compared and evaluated before a decision concerning identification was reached.

[Fisher 89] presented a model invocation method which integrates evidence from object descriptions acquired from the image, and relationship evidence from class and structural associations from the model library. The process helps to build strong hypotheses to be further verified by making evidence available at an early stage of recognition, a consequence of it being that the cost of hypothesis making becomes high. A network model is used for integrating all the evidence, and node values are computed using harmonic means and threshold limits.

An accumulator based method is presented by [Shapiro.L.G. & Lu 90] for determining the view class that best matches an unknown view of the object. The method is based in assigning a Gaussian distribution evidence value to accumulator lists for the view classes, and then multiplying them by weights previously defined according to empirical tests. The method was proposed to deal with inexact matching in a context where features in a particular view of an object are described using relational pyramid structures.

[Ettinger 87] addressed model library indexing of 2-D objects using a scheme where models are linked together by the presence of similar 2-D parts. The objects used, images of traffic signs, are described as assemblies of “Curvature Primal Sketches”. After feature extraction hypotheses are ordered by application of a Hough Transform, an Interpretation Tree search process is run on the indexed library to find the best matches.

[Flynn & Jain 91] presented a constraint based search recognition system for 3-D objects. Objects are described using CAD model definitions, for which special constraints are designed for the Interpretation Tree matching algorithm. Surface patches are ordered before creating the Interpretation Tree according to patch area.

[Grimson 88] proposed an extension of the Interpretation Tree matching algorithm for dealing with 2-D objects, parameterized according to rotational, translation, and scaling degrees of freedom. Objects are represented by their boundaries, and matching proceeds by searching the Interpretation Tree for isolated parts. Parts segmentation is not addressed. Specific geometric constraints for the case of a 2-D articulated object (a pair of scissors) are presented.

5.10 Chapter Summary

This chapter has presented an efficient and robust method to perform matching of 3-D objects from range image data. The input features are provided by the segmentation and classification methods described in Chapters 3 and 4 respectively.

The method divides the problem into two stages: an indexing stage and a constrained search stage with fuzzy degrees of similarity. The indexing stage selects a subset of the model library to be fully searched based on the output of the geon classifier (Chapter 4). The models are indexed in the library by grouping them on the common part primitives (geons) they share. This organization allows for robustness in recognition since a meaningful and reliable classification relates these parts to their parameters. This indexing scheme has also the property of sublinear growth with the size of the library, and so it does not compromise its performance when new models are added in the database.

A quality measure of similarity is introduced in a constrained interpretation tree based search algorithm providing a robust and efficient mechanism to rank and compare inexact matchings. The quality measure is built by using the Theory of Fuzzy Sets [Zadeh 78], first by evaluating feature bindings (D, M) (Data primitives, Model primitives) with fuzzy predicates, and deriving a fuzzy degree of similarity which ranks the valid hypotheses reached in the constrained search.

Results are given for a variety of complex 3-D shapes, and identification and ranking are successfully achieved. The next chapter presents a method to estimate the pose in 3-D of articulated objects, as applied to the objects identified in this stage.

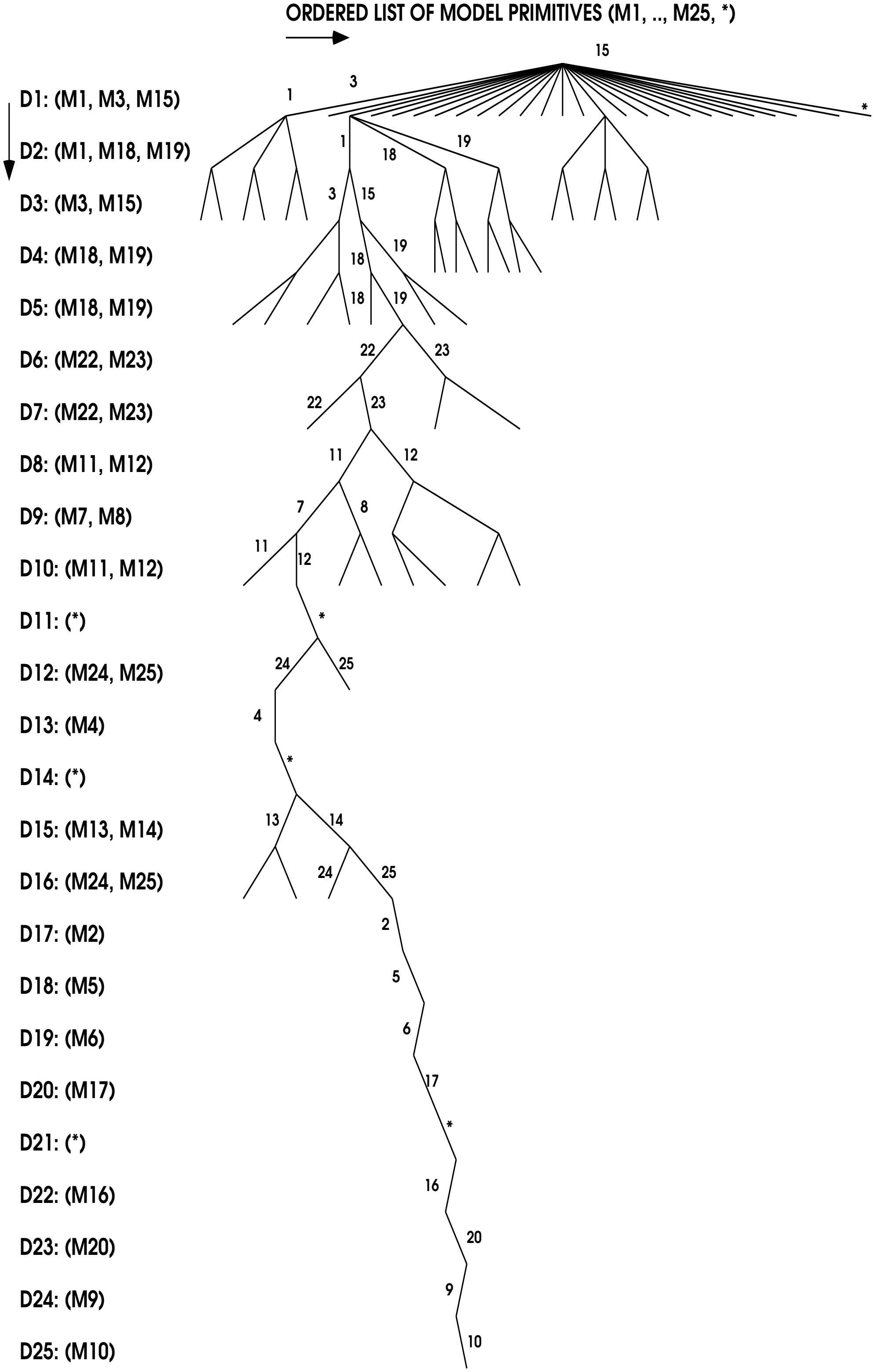


Figure 5.11: Portion of the Interpretation Tree for object "doll" showing the successful path and branches explored at each level.

Chapter 6

Pose Estimation of Constrained Non-Rigid 3-D Objects using Kalman Filtering

This chapter is concerned with the pose estimation of 3-D objects that have partial constraints between some of the parts such as articulations. This is a necessary step to locate the successfully matched hypotheses achieved in Chapter 5. The localization of the hypotheses is then used to overlay the model onto the original 3-D data, and, in this case, verify the match.

Pose estimation of 3-D constrained objects is a difficult and important problem for 3-D model based object recognition. The method described in this chapter is based on the Kalman Filtering framework, which is a general technique for state and parameter estimation of stochastic systems [Maybeck 79].

This chapter is organized as follows. Section 6.1 states the general 3-D to 3-D pose estimation problem, Section 6.2 describes the Kalman filtering methodology and its specific formulation developed for the problem presented here, Section 6.3 presents results on applying the method on matched models and data from Chapter 5, Section 6.4 discusses the results achieved and the applicability of the method, related work is reviewed in Section 6.5, and Section 6.6 summarizes the chapter.

6.1 Localization of the Valid Hypotheses

A crucial problem in model-based recognition is to find the attitude, or pose, in the 3-D space of the models matched from the database. It involves computing object position and orientation relative to a camera reference frame, *i.e.* finding a transformation in 3-D that maps the model onto the data image.

The task of computing the transformation to locate the object depends on the dimensionality of the data from the sensor, of the models, and of the world or coordinate space considered. In this thesis, the sensor data, the models, and the world are all three-dimensional. The mapping therefore is a 3-D rigid transformation, and it has 6 degrees of freedom.

Complicating the problem is the additional need to estimate internal positional degrees of freedom in the case of complex parts with articulations. Satisfaction of these constraints in the final solution is commonly achieved by a separate process, in which the uncertainty on the data positions is not represented uniformly.

We define next the 3-D to 3-D pose estimation problem we are concerned with in this work.

6.1.1 The General 3-D to 3-D Pose Estimation Problem

Let $\mathcal{M}$ be a model of a 3-D object, which is represented by n features $\mathbf{mf}$. The position $\mathbf{p}_{mf}$ and orientation $\mathbf{o}_{mf}$ of the features are assumed to be known relative to an object-centred frame, and the features are represented as:

$$\mathbf{mf}_i \quad = \quad (\mathbf{p}_{mf_i}, \mathbf{o}_{mf_i}) \qquad i = 1, ..., n \tag{6.1}$$

Let $\mathcal{M}'$ be a measurement of a 3-D object, which was acquired from the scene data and consists of m features $\hat{\mathbf{sf}}$. The position $\hat{\mathbf{p}}_{sf}$ and orientation $\hat{\mathbf{o}}_{sf}$ of the features are assumed to be known relative to a viewer-centred reference frame, and the features are represented as:

$$\hat{\mathbf{sf}}_j \quad = \quad (\hat{\mathbf{p}}_{sf_j}, \hat{\mathbf{o}}_{sf_j}) \qquad j = 1, ..., m \tag{6.2}$$

The 3-D to 3-D pose estimation problem is to find a transformation $\mathbf{T}(\mathbf{R},\mathbf{t})$, with rotation $\mathbf{R}$ and translation $\mathbf{t}$ components, that maps the coordinates of the features $\mathbf{mf}_i$ (object-centred) onto the corresponding coordinates of the features $\hat{\mathbf{sf}}_{j(i)}$ (viewer-centred). If the features $\hat{\mathbf{sf}}_{j(i)}$ and $\mathbf{mf}_i$ are correctly matched, and there is no noise in the measurements one should have:

$$\begin{array}{rcl} \mathbf{mf}_i & \overset{\mathbf{T(R,t)}}{\longrightarrow} & \hat{\mathbf{sf}}_{j(i)} \qquad i = 1, ..., l \qquad l \leq min(n,m) \\ \mathbf{R}.\mathbf{p}_{mf_i} + \mathbf{t} & = & \hat{\mathbf{p}}_{sf_i} \\ \mathbf{R}.\mathbf{o}_{mf_i} & = & \hat{\mathbf{o}}_{sf_i} \end{array} \tag{6.3}$$

In normal conditions noise in the measurements is always present, and it comes from a variety of sources. In this thesis, noise is considered in the pose estimation process, and it is assumed to have a Gaussian distribution. Although it is not always true that the noise is Gaussian this choice is a good approximation given the numerous sources of noise and the difficulty in defining all of them. It can be represented by an uncertainty matrix associated with the measured features,

$$\mathbf{sf} \;=\; [\hat{\mathbf{sf}}, \Lambda] \tag{6.4}$$

where $\mathbf{sf}$ is the location vector of the features, $\hat{\mathbf{sf}}$ is the measured location vector, and Λ is the uncertainty matrix associated with the measurements. This uncertainty is represented in fact by a covariance matrix, which, besides being a compact and efficient representation for uncertainty, facilitates fusion of measurements.

6.1.2 Parameterized and Articulated Objects

A parameterized or constrained model of a 3-D object is a description which imposes general constraints between the model components. These constraints can define equality relations such as: co-linearity, co-planarity, angle relationships, and edge length relationships; or they can define inequality relations such as range of distances or angles. These are typical parameterizations on positions, other types of parameterizations can also involve shape functions.

An articulated object is a particular type of parameterized object having constraints on the spatial locations of the parts.

3-D to 3-D Pose Estimation with Parameterized Objects

Let $\mathcal{M}$ be a parameterized model of a 3-D object, which consists of a set of rigid parts or components P,

$$\mathcal{M} \;=\; \{P_i\} \qquad i = 1, ..., n \tag{6.5}$$

Each part P_i has a location vector $\mathbf{mf}_i$, which consists of position and orientation in 3-D relative to an object-centred frame. The part P_i in our case is a rigid object, and its pose will be then estimated as such, however the model $\mathcal{M}$ which is an assembly of parts P_i is a non-rigid object. This identifies therefore two distinct stages in the pose estimation of these objects. First, the pose of each part is estimated where the constraints on the object $\mathcal{M}$, which are relationships between these parts, are not considered. After that, the pose of the complete object is estimated considering the already available poses for each part and particularly dealing with enforcing the constraints specified between some of the parts.

Let $\mathcal{M}'$ be the object found in the matching to be correspondent to $\mathcal{M}$. $\mathcal{M}'$ can be expressed as:

$$\mathcal{M}' \;=\; \{P_i'\} \qquad i = 1, ..., n \tag{6.6}$$

The set of constraints Ψ existing between the parts is defined as relationships between the correspondent transformation vectors of the parts, and is represented as:

$$\Psi \;=\; \{\Psi_j(\mathbf{T}_l, \mathbf{T}_k) = 0\} \tag{6.7}$$

The next section presents the Kalman Filtering technique, which is a general and robust tool for parameter estimation. We propose to use this technique to deal with our pose estimation problem, and an algorithm is developed and results are shown further in this chapter. Because of its simple formulation, and capability to integrate in a consistent way a variety of constraints and uncertainty about the data, we found it to be the right framework for our problem of pose estimation of constrained non-rigid 3-D objects.

6.2 The Kalman Filtering Framework

This section describes the Kalman filtering framework, which is an optimal parameter estimation technique for stochastic systems. The basic Kalman filtering concepts and equations are described first. The basic equations hold only for cases where the model of the system is linear. A way to overcome this limitation is to apply the Extended Kalman Filtering derivations. These derivations, and particularly the algorithm for an Iterated Extended Kalman Filter are shown following the linear case. The 3-D to 3-D pose estimation problem can be formulated under this framework as will be shown at the end of this section.

6.2.1 The Basic Kalman Filter

The Kalman Filter is a technique to estimate a parameter vector of a stochastic linear system from a sequence of measurements. In general the parameter vector, or state vector, is dynamic, *i.e.* it changes with time. However, in this application (3-D pose estimation) the state vector is static and the description that follows considers only the static case.

Figure 6.1 shows a block diagram of a basic Kalman Filter. The equations will only be stated here. For a complete derivation of the Kalman Filter equations a good reference is [Maybeck 79].

Let $\mathbf{a}$ be a state of a system, and $\mathbf{x}_i$ a measurement of the state. A linear sensor model or measurement equation relating $\mathbf{a}$ and $\mathbf{x}_i$ is:

$$\mathbf{x}_i = \mathbf{M}_i\mathbf{a} \tag{6.8}$$

where $\mathbf{M}_i$ is a $m \times n$ matrix at step i. The Kalman Filter will process at each step i an imperfect measurement with mean $\hat{\mathbf{x}}_i$, and covariance Λ_i, according to the model in equation (6.8), and produce a new estimate of the state with mean $\hat{\mathbf{a}}_i$, and covariance Σ_i.

It is assumed that the noise ϵ_i, which relates $\hat{\mathbf{x}}_i = \mathbf{x}_i + \epsilon_i$ satisfies:

$$E[\epsilon_i] = 0 \;\; ; \;\; Var[\epsilon_i] = \Lambda_i \;\; ; \;\; Covar[\epsilon_i, \epsilon_j] = 0 \;\; for \;\; i \neq j.$$

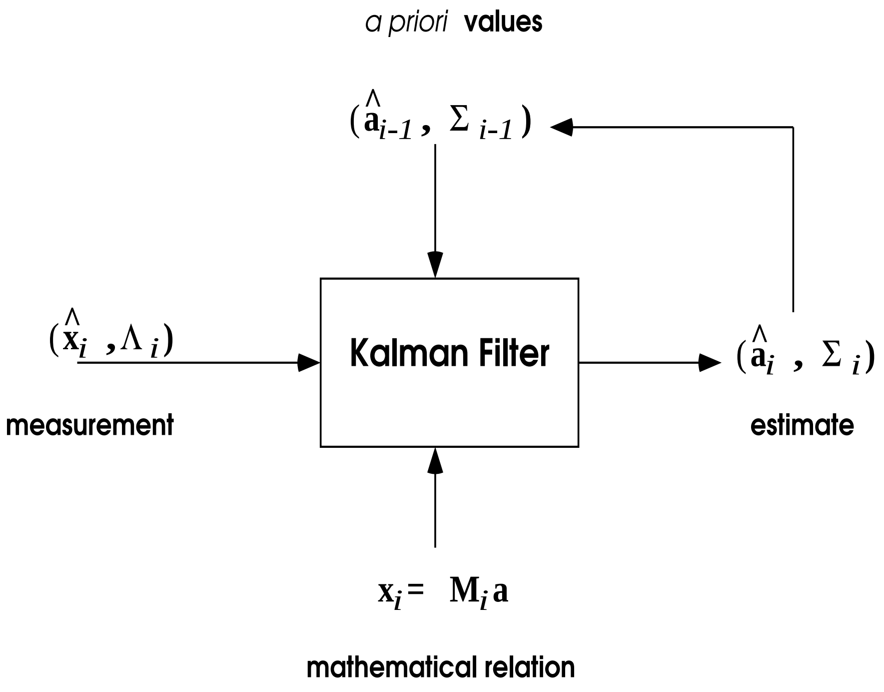


Figure 6.1: Block diagram of the basic Kalman Filter.

The equations of the basic Kalman Filter are then:

$$\hat{\mathbf{a}}_i = \hat{\mathbf{a}}_{i-1} + \mathbf{K}_i(\hat{\mathbf{x}}_i - \mathbf{M}_i\hat{\mathbf{a}}_{i-1}) \tag{6.9}$$

$$\Sigma_i = (\mathbf{I} - \mathbf{K}_i\mathbf{M}_i)\Sigma_{i-1} \tag{6.10}$$

$$\mathbf{K}_i = \Sigma_{i-1}\mathbf{M}_i^t(\Lambda_i + \mathbf{M}_i\Sigma_{i-1}\mathbf{M}_i^t)^{-1} \tag{6.11}$$

The equations (6.9, 6.10, 6.11) represent the state estimate update, the state covariance update, and the Kalman gain matrix respectively. The Kalman Filter in any case will need at each time step i three inputs:

1. *a priori* estimate values $(\hat{\mathbf{a}}_{i-1}, \Sigma_{i-1})$.
2. measurement input $(\hat{\mathbf{x}}_i, \Lambda_i)$.
3. measurement model $\mathbf{x}_i = \mathbf{M}_i\mathbf{a}$.

For most practical cases the measurement model will not be linear. However, if the model can be approximately linearized the Kalman Filter framework can still be applied. The next section deals with this extension.

6.2.2 The Iterated Extended Kalman Filter

The basic Kalman Filter as shown in Section 6.2.1 is applicable to linear systems only. In general, and particularly in the problem addressed here, the measurement equations are non-linear implicit functions of the type

$$f_i(\mathbf{x}_i, \mathbf{a}) \quad = \quad 0 \tag{6.12}$$

In such a case the solution is to assume that the means $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{a}}_{i-1}$ are good approximations to $\mathbf{x}_i$ and $\mathbf{a}$, and compute f_i by expanding it in a Taylor series around $(\hat{\mathbf{x}}_i, \hat{\mathbf{a}}_{i-1})$:

$$f_i(\mathbf{x}_i, \mathbf{a}) \quad = \quad 0 \approx f_i(\hat{\mathbf{x}}_i, \hat{\mathbf{a}}_{i-1}) + \frac{\partial f_i}{\partial \mathbf{x}_i}(\mathbf{x}_i - \hat{\mathbf{x}}_i) + \frac{\partial f_i}{\partial \mathbf{a}}(\mathbf{a} - \hat{\mathbf{a}}_{i-1}) \tag{6.13}$$

where only the first order terms are kept.

The Jacobians

$$\frac{\partial f_i}{\partial \mathbf{x}_i} \quad = \quad [\nabla_{\mathbf{x}_i} f_i^t]^t \tag{6.14}$$

$$\frac{\partial f_i}{\partial \mathbf{a}} \quad = \quad [\nabla_{\mathbf{a}} f_i^t]^t \tag{6.15}$$

are evaluated at $(\hat{\mathbf{x}}_i, \hat{\mathbf{a}}_{i-1})$. The measurement equation can now be written as

$$\mathbf{y}_i \quad = \quad \mathbf{M}_i \mathbf{a} + \eta_i \tag{6.16}$$

where

$$\mathbf{y}_i \quad = \quad -f_i(\hat{\mathbf{x}}_i, \hat{\mathbf{a}}_{i-1}) + \frac{\partial f_i}{\partial \mathbf{a}} \hat{\mathbf{a}}_{i-1} \tag{6.17}$$

$$\mathbf{M}_i \quad = \quad \frac{\partial f_i}{\partial \mathbf{a}} \tag{6.18}$$

$$\eta_i \quad = \quad \frac{\partial f_i}{\partial \mathbf{x}_i}(\mathbf{x}_i - \hat{\mathbf{x}}_i) \tag{6.19}$$

$\mathbf{y}_i$ represents the new measurement, $\mathbf{M}_i$ is the linear transformation between the measurement and the state vector. The term η_i denotes the random noise in the measurement, and satisfies the following:

$$E[\eta_i] \quad = \quad 0 \tag{6.20}$$

$$\mathbf{W}_i \equiv Var[\eta_i] \quad = \quad \frac{\partial f_i}{\partial \mathbf{x}_i} \Lambda_i \left(\frac{\partial f_i}{\partial \mathbf{x}_i}\right)^t \tag{6.21}$$

$$Covar[\eta_i, \eta_j] \quad = \quad 0 \quad \forall i \neq j \tag{6.22}$$

The Extended Kalman Filter (EKF) equations as a result of the linearization are:

$$\hat{\mathbf{a}}_i = \hat{\mathbf{a}}_{i-1} + \mathbf{K}_i(\mathbf{y}_i - \mathbf{M}_i\hat{\mathbf{a}}_{i-1}) = \hat{\mathbf{a}}_{i-1} - \mathbf{K}_i f_i(\hat{\mathbf{x}}_i, \hat{\mathbf{a}}_{i-1}) \tag{6.23}$$

$$\Sigma_i = (\mathbf{I} - \mathbf{K}_i\mathbf{M}_i)\Sigma_{i-1} \tag{6.24}$$

or equivalently

$$\Sigma_i^{-1} = \Sigma_{i-1}^{-1} + \mathbf{M}_i^t\mathbf{W}_i^{-1}\mathbf{M}_i \tag{6.25}$$

$$\mathbf{K}_i = \Sigma_{i-1}^{-1}\mathbf{M}_i^t(\mathbf{W}_i + \mathbf{M}_i\Sigma_{i-1}\mathbf{M}_i^t)^{-1} \tag{6.26}$$

The linearization process of f_i can be misleading if $\hat{\mathbf{a}}_{i-1}$ is not very close to the true value of $\mathbf{a}$. To reduce the influence of these errors local iterations are applied, updating $\hat{\mathbf{a}}_i$, re-linearizing f_i with the new $\hat{\mathbf{a}}_i$, and computing $\mathbf{K}_i$, $\mathbf{y}_i$, and $\mathbf{M}_i$ until $\hat{\mathbf{a}}_i^{(k)} - \hat{\mathbf{a}}_i^{(k-1)}$ is close to zero. This new filter is called Iterated Extended Kalman Filter (IEKF) [Maybeck 79], and it is this formulation that will be used for the problem of pose estimation.

6.3 Applying the IEKF to Estimate Pose of Constrained 3-D Volumetric Objects

This section describes a method to estimate the pose of 3-D objects using the Kalman Filter. We are concerned here particularly with part based objects such as the ones dealt with throughout this thesis. However, the solution can be extended to deal with other object representations.

6.3.1 Estimating the Initial Transformation Vector

In order to formulate the problem in the IEKF framework it is necessary to identify:

- The state or the vector of parameters to be estimated.
- The measurements or observations.
- The measurement model.

The State or Parameters to be Estimated

For each feature or part P_i the parameter vector to be estimated consists of the transformation vector $\mathbf{T}_i(\mathbf{R}_i, \mathbf{t}_i)$. The translation component is represented by vector $\mathbf{t}_i$

$$\mathbf{t}_i \quad = \quad (t_x, t_y, t_z)^t \tag{6.27}$$

and the rotation component by quaternion $\tilde{\mathbf{q}}_i$ [Horn 86]:

$$\tilde{\mathbf{q}}_i \quad = (q_0, \mathbf{q}) = \quad (q_0, q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k})$$

Quaternion $\tilde{\mathbf{q}}_i$ has to satisfy the constraint that $\|\tilde{\mathbf{q}}_i\|^2 = 1$. This constraint is enforced with the use of an additional measurement fused in the filter, as will be described in the "measurement model".

The transformation $\mathbf{T}_i$ is a seven dimensional vector as:

$$\mathbf{T}_i \quad = \quad \begin{pmatrix} \tilde{\mathbf{q}}_i \\ \mathbf{t}_i \end{pmatrix}$$

$\mathbf{T}_i$ represents a transformation for the corresponding part $\mathcal{P}_i$. The model $\mathcal{M}$ is an assembly of the parts P_i, as in equation (6.6).

The vector $\mathbf{T}$ which includes the transformations of the parts P_i for the model $\mathcal{M}$ is then represented by the following:

$$\mathbf{T}_{\mathcal{M}} \quad = \begin{pmatrix} \mathbf{T}_1(\mathbf{t}_1, \tilde{\mathbf{q}}_1) \\ \mathbf{T}_2(\mathbf{t}_2, \tilde{\mathbf{q}}_2) \\ . \\ . \\ \mathbf{T}_n(\mathbf{t}_n, \tilde{\mathbf{q}}_n) \end{pmatrix} \qquad \textit{and} \qquad \Sigma_{\mathcal{M}} = \begin{pmatrix} \Sigma_1 \\ \Sigma_2 \\ . \\ . \\ \Sigma_n \end{pmatrix} \tag{6.28}$$

The Observations

The measurements for each volumetric part P_i of the model will be point positions. This simplifies the measurement model; however, in order to constrain both the rotation and the translation components of the transformation three non-colinear points will have to be processed. From the superquadrics representing each part we propose to use the center of gravity point **gp**, the extreme point at the major axis **map**, and the extreme point at the minor axis **mip**. Figure 6.2 shows a superquadric with these points labelled for visualization.

Let $\mathbf{u}_i \in \{\mathbf{gp}_i, \mathbf{map}_i, \mathbf{mip}_i\}$ be a point in the part model (object-centred frame). Let $\mathbf{u'}_i \in \{\mathbf{gp'}_i, \mathbf{map'}_i, \mathbf{mip'}_i\}$ be a point in the part data (viewer-centred frame). The observations, or the measurements of these points will then be:

$$\hat{\mathbf{u}}'_i \;=\; \mathbf{u'}_i + \epsilon_i$$

where ϵ_i represents the noise and satisfies:

$$E[\epsilon_i] = 0 \;\;;\;\; Var[\epsilon_i] = \Lambda_i \;\;;\;\; Covar[\epsilon_i, \epsilon_j] = 0 \;\; for \;\; i \neq j.$$

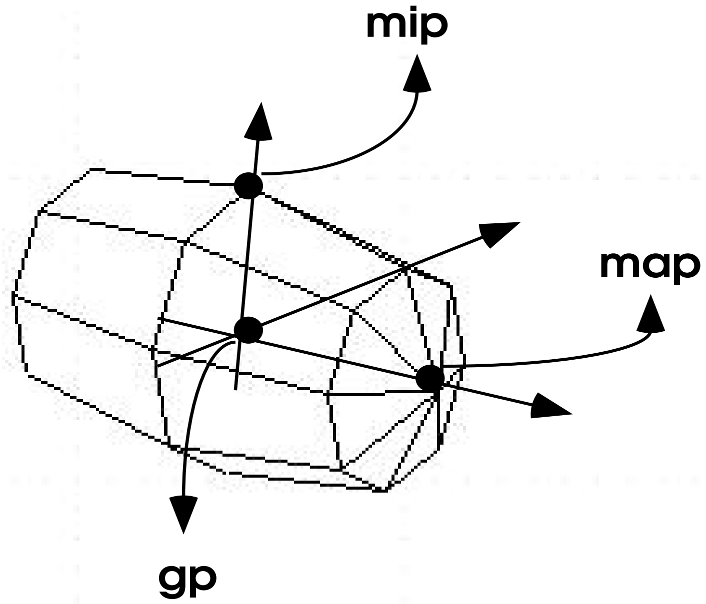


Figure 6.2: Superquadric model showing the object-centred reference frame with the points **gp**, **map**, and **mip**.

The Measurement Model

Representing the rotation as a quaternion, the measurement equation relating the state and the observations, for each feature point i is:

$$\tilde{\mathbf{u}}'_i \;=\; \tilde{\mathbf{q}}_i \tilde{\mathbf{u}}_i \tilde{\mathbf{q}}_i^* + \tilde{\mathbf{t}}_i \tag{6.29}$$

where $\tilde{\mathbf{u}}'_i, \tilde{\mathbf{u}}_i, \tilde{\mathbf{t}}_i$ are quaternion representations of the respective vectors $\mathbf{u'}_i, \mathbf{u}_i, \mathbf{t}_i$. Multiplying equation (6.29) by $\tilde{\mathbf{q}}_i$, and noticing that $\tilde{\mathbf{q}}_i \tilde{\mathbf{q}}_i^* = 1$ yields (see Appendix E for definitions and operations using quaternions):

$$\tilde{\mathbf{u}'}_i\tilde{\mathbf{q}}_i \quad = \quad \tilde{\mathbf{q}}_i\tilde{\mathbf{u}}_i + \tilde{\mathbf{t}}_i\tilde{\mathbf{q}}_i \tag{6.30}$$

Equation (6.30) can be transformed into a suitable measurement equation by isolating the vector component of it, and dividing it by q_0 [Hel-Or 93], which yields:

$$f_i(\mathbf{u}_i, \mathbf{u'}_i, \mathbf{T}_i) \quad \equiv \langle \mathbf{u}_i + \mathbf{u'}_i \rangle \mathbf{s}_i + (\mathbf{u'}_i - \mathbf{u}_i) - (\mathbf{I}_{3\times3} - \langle \mathbf{s}_i \rangle)\mathbf{t}_i = \quad 0 \tag{6.31}$$

where $\mathbf{s}$ is a new vector defined as $\mathbf{s} = \mathbf{q}/q_0$, $\mathbf{I}_{3\times3}$ is the 3×3 identity matrix, and the operator $\langle . \rangle$ represents the matrix form of a cross product,

$$\langle \mathbf{w} \rangle \equiv \langle (w_1, w_2, w_3) \rangle \quad = \begin{pmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{pmatrix} \ ; \ \langle \mathbf{w} \rangle \mathbf{v} = -\langle \mathbf{v} \rangle \mathbf{w} = \mathbf{w} \times \mathbf{v}$$

Equation (6.31) is a non-linear relationship, and by using an IEKF as our estimation tool the following linearization of eq. (6.31) around $(\hat{\mathbf{T}}_i^{(k-1)}, \hat{\mathbf{u}}'^{(k)}_i)$ is applied:

$$\begin{aligned} f_i^{(k)}(\mathbf{u}_i^{(k)}, \mathbf{u}'^{(k)}_i, \mathbf{T}_i) = 0 \quad \approx \quad & f^{(k)_i}(\mathbf{u}_i^{(k)}, \hat{\mathbf{u}}'^{(k)}_i, \hat{\mathbf{T}}_i^{(k-1)}) + \frac{\partial f_i^{(k)}}{\partial \mathbf{u}'^{(k)}_i}(\mathbf{u}'^{(k)}_i - \hat{\mathbf{u}}'^{(k)}_i) + \\ & \frac{\partial f_i^{(k)}}{\partial \mathbf{T}_i}(\mathbf{T}_i - \hat{\mathbf{T}}_i^{(k-1)})) \end{aligned} \tag{6.32}$$

where index k indicates the general k^{th} step of the filter process.

Developing (6.32) yields:

$$\begin{aligned} \langle \hat{\mathbf{s}}_i^{(k-1)} \rangle \hat{\mathbf{t}}_i^{(k-1)} + \mathbf{u}_i^{(k)} - \hat{\mathbf{u}}'^{(k)}_i \quad \approx \quad & [\langle \hat{\mathbf{u}}'^{(k)}_i + \mathbf{u}_i^{(k)} - \hat{\mathbf{t}}_i^{(k-1)} \rangle, (\langle \hat{\mathbf{s}}_i^{(k-1)} \rangle - \mathbf{I}_{3\times3})]\mathbf{T}_i + \\ & [\mathbf{I}_{3\times3} - \langle \hat{\mathbf{s}}_i^{(k-1)} \rangle](\mathbf{u}'^{(k)}_i - \hat{\mathbf{u}}'^{(k)}) \end{aligned} \tag{6.33}$$

Writing equation (6.33) in a linear form:

$$\mathbf{y}_i^{(k)} \quad = \quad \mathbf{M}_i^{(k)}\mathbf{T}_i + \eta_i^{(k)} \tag{6.34}$$

where

$$\begin{aligned} \mathbf{y}_i^{(k)} \quad &= \quad \langle \hat{\mathbf{s}}_i^{(k-1)} \rangle \hat{\mathbf{t}}_i^{(k-1)} + \mathbf{u}_i^{(k)} - \hat{\mathbf{u}}'^{(k)}_i \\ \mathbf{M}_i^{(k)} \quad &= \quad [\langle \hat{\mathbf{u}}'^{(k)}_i + \mathbf{u}_i^{(k)} - \hat{\mathbf{t}}_i^{(k-1)} \rangle, (\langle \hat{\mathbf{s}}_i^{(k-1)} \rangle - \mathbf{I}_{3\times3})] \\ \eta_i^{(k)} \quad &= \quad [\mathbf{I}_{3\times3} - \langle \hat{\mathbf{s}}_i^{(k-1)} \rangle](\mathbf{u}'^{(k)}_i - \hat{\mathbf{u}}'^{(k)}_i) \end{aligned} \tag{6.35}$$

$\mathbf{y}_i^{(k)}$ is the new measurement, $\mathbf{M}_i^{(k)}$ is the new linear relationship between $\mathbf{y}_i^{(k)}$ and $\mathbf{T}_i$, and $\eta_i^{(k)}$ represents the noise in the measurement.

To enforce the quaternion constraint $\|\tilde{\mathbf{q}}\|$ for it to represent the rotation, an additional measurement is derived and fused in the filter with uncertainty equal to zero. The measurement to be concatenated in the equations (6.36), is:

$$1 + (\hat{\mathbf{T}}_i^{(k-1)})^t \begin{bmatrix} \mathbf{I}_{4\times 4} & 0 \\ 0 & 0 \end{bmatrix} \hat{\mathbf{T}}_i^{(k-1)} \tag{6.36}$$

The initial conditions for the IEKF, step 0, are set to be a random chosen value with infinity uncertainty. This means to say that no *a priori* knowledge is assumed for this stage of the estimation. From the simulations and experiments with real data we found that this choice did not affect the final results. The initial transformation vector consists of the collection of the transformations computed separately for each part of the model. At this stage the constraints between the parts are not taken into consideration. For each part the pose is computed by applying the IEKF equations as given by the model in equation (6.36). The poses or state parameters $\mathbf{T}_i$, for parts P_i, form the initial transformation vector, equation (6.28). The next section shows how to add the constraints in the IEKF computation.

6.3.2 Adding Constraints as Perfect Measurements

In the second stage of the algorithm the initial transformation vector together with its uncertainty matrix, eq. (6.28), are used as the *a priori* estimates in another IEKF computation. The constraints are now added as perfect artificial measurements, *i.e.* with no uncertainty, in order to be integrated in the final pose of the object. This section describes the formulation and linearization of these constraints. In the models only one type of constraint is included, a simple constraint for articulation and it is described together with its linearized form for the filter .

In order to satisfy the constraints specified in the model they are linearized and included in the IEKF as additional measurements having zero uncertainty. The set of constraints Ψ on the vector $\mathbf{T}$ will be:

$$\{\Psi_j(\mathbf{T}) = 0\} \quad j = 1, ..., m$$

The linearization of a given constraint $\Psi_j(\mathbf{T}) = 0$ is performed around the measurement

$\hat{\mathbf{u}}'$ as:

$$\Psi_j(\mathbf{T}) \quad = 0 \approx \Psi_j(\hat{\mathbf{u}}') + \tfrac{\partial \Psi_j}{\partial \mathbf{T}}(\mathbf{T} - \hat{\mathbf{u}}') \tag{6.37}$$

Considering it in a linear form, and without noise:

$$\mathbf{y}_j \quad = \quad M_j \mathbf{T} \tag{6.38}$$

where,

$$\mathbf{y}_j \quad = \quad -\Psi_j(\hat{\mathbf{u}}') + \left(\frac{\partial \Psi_j}{\partial \mathbf{T}} \right) \hat{\mathbf{u}}' \tag{6.39}$$

$$M_j \quad = \quad \frac{\partial \Psi_j}{\partial \mathbf{T}} \tag{6.40}$$

Appending all the constraints in the vector form:

$$\mathbf{y} \quad = \quad M\mathbf{T} \tag{6.41}$$

where,

$$\mathbf{y} \;=\; \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ . \\ . \\ \mathbf{y}_m \end{pmatrix} \qquad \mathit{and} \qquad M \;=\; \begin{pmatrix} M_1 \\ M_2 \\ . \\ . \\ M_m \end{pmatrix} \tag{6.42}$$

The estimate for $\mathbf{T}$ from the IEKF will be:

$$\hat{\mathbf{T}} \quad = \quad \hat{\mathbf{T}}^- + \Lambda M^t (M \Lambda M^t)^{-1} (\mathbf{y} - M\hat{\mathbf{T}}^-) \tag{6.43}$$

A Simple Constraint for Articulation

One simple constraint imposed in the objects tested is that the Euclidean distance between the parts is kept constant, which can be formulated as the following additional measurement:

$$\Psi_j(\mathbf{t}_l, \mathbf{t}_k) \quad = \|\mathbf{t}_k - \mathbf{t}_l\|^2 - \xi^2(l,k) = \quad 0 \tag{6.44}$$

where $\xi(k,l)$ is the Euclidean 3-D distance between the two adjacent parts P_k and P_l, and is imposed as a constant in this case.

Linearizing equation (6.44) around the two measurement points $(\hat{\mathbf{u}}'_l, \hat{\mathbf{u}}'_k)$ yields:

$$\frac{1}{2}(\xi^2(l,k) + (\hat{\mathbf{u}}'_l - \hat{\mathbf{u}}'_k)^2) \quad = \quad (\hat{\mathbf{u}}'_l - \hat{\mathbf{u}}'_k)(\mathbf{t}_l - \mathbf{t}_k) \tag{6.45}$$

The vectors $\mathbf{t}_l$ and $\mathbf{t}_k$ are the translation components of the transformation only. The measurement points $(\hat{\mathbf{u}}'_l, \hat{\mathbf{u}}'_k)$ are chosen to be the centres of the superquadric object-centred reference frame, *i.e.* points $\mathbf{gp}$ for parts l and k respectively. This type of constraint for articulation approximates well the allowable movements normally found in the objects and models we are dealing with.

By concatenating all constraints of the type (6.45) the measurement vectors as in equation (6.42) are constructed. This measurement vector associated with uncertainty matrix zero, *i.e.* no noise, are fused in a second IEKF estimation where the *a priori* estimates are the vectors of transformations for the rigid parts. This computational framework provides an integrated and robust method to estimate poses of parameterized 3-D objects under uncertainty. The next section summarizes the complete algorithm. Experiments and results are shown in Section 6.4.

6.3.3 Estimating the Pose by Fusing the Measurements in Batch Mode

One way to include the constraints into the IEKF is by considering them as artificial measurements with zero uncertainty, and supplying them as one enlarged measurement matrix, *i.e.* in a batch mode.

The complete pose estimation process can be summarized in the following steps:

1. For each part P_i of the model, apply the IEKF (equations (6.36)) to estimate $\mathbf{T}_i$ by using the three characteristic points $(\mathbf{gp}_i, \mathbf{map}_i, \mathbf{mip}_i)$. For a model $\mathcal{M} = \{P_i\}_{i=1,\ldots,n}$, equation (6.28). Where the initial conditions are set to no *a priori* knowledge, infinity uncertainty matrix.
2. Construct $(\hat{\mathbf{T}}_{\mathcal{M}}, \Sigma_{\mathcal{M}})$ as the *a priori* estimates for the IEKF.
3. Construct vector $\mathbf{y}$, as from right side of equation (6.45), of measurements with all the constraints having zero uncertainty.

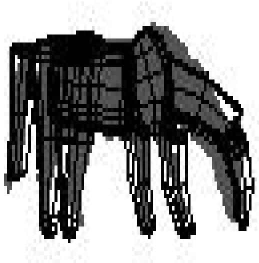


Figure 6.3: View 1 of model for object "horse" shown overlayed on 3-D data in the final estimated position. This is the viewpoint from which the range image was obtained.

4. Construct model relation vector **M**, from left side of equation (6.45), as a concatenation of all the constraints linearized.
5. Fuse inputs from steps 2, 3, and 4 into the IEKF in batch mode, equation (6.43).

6.4 Experiments and Results

This section presents results on applying the IEKF to estimate the pose of the matched models found in Chapter 5, and simulation experiments on artificially generated data.

Matched models are shown in the final estimated position from three different views overlayed on the original 3-D data points: The original viewpoint from which the range is obtained, a view from the top of the object, and a lateral view. Figures 6.3-6.5, 6.6-6.8, 6.9-6.11, and 6.12-6.14 are results for the objects "horse", "cow", "giraffe", and "doll" respectively. In these figures the final matched model is shown overlayed on the 3-D data, whereas in Chapter 5 the overlayed figures showed the estimated superquadric data descriptions (after segmentation) on the 3-D data.

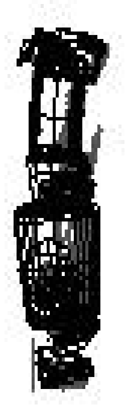


Figure 6.4: View 2 (from above) of model for object “horse” shown overlayed on 3-D data in the final estimated position.

Figure 6.5: View 3 of model for object “horse” shown overlayed on 3-D data in the final estimated position.

Figure 6.6: View 1 of model for object “cow” shown overlayed on 3-D data in the final estimated position. This is the viewpoint from which the range image was obtained.

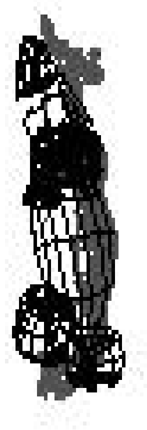


Figure 6.7: View 2 (from above) of model for object “cow” shown overlayed on 3-D data in the final estimated position.

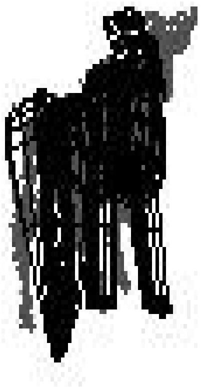


Figure 6.8: View 3 of model for object "cow" shown overlayed on 3-D data in the final estimated position.

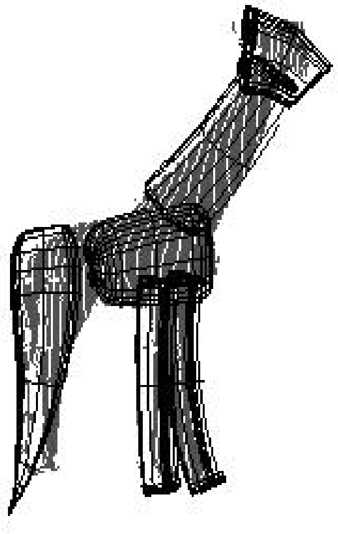


Figure 6.9: View 1 of model for object "giraffe" shown overlayed on 3-D data in the final estimated position. This is the viewpoint from which the range image was obtained.

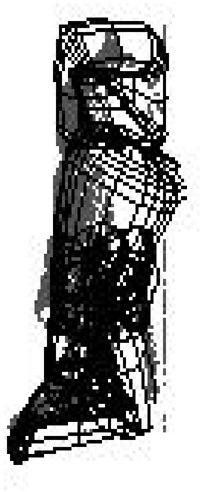


Figure 6.10: View 2 (from above) of model for object "giraffe" shown overlayed on 3-D data in the final estimated position.

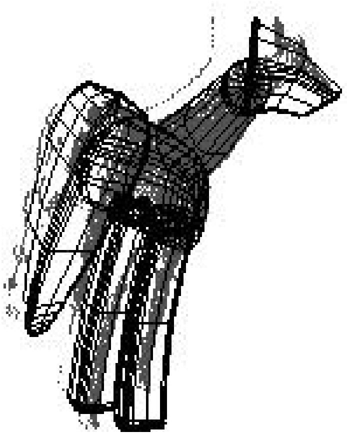


Figure 6.11: View 3 of model for object "giraffe" shown overlayed on 3-D data in the final estimated position.

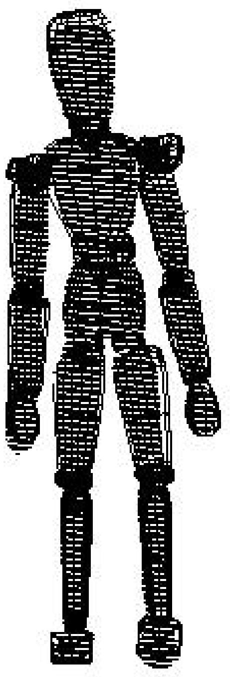


Figure 6.12: View 1 of model for object “doll” shown overlayed on 3-D data in the final estimated position. This is the viewpoint from which the range image was obtained.

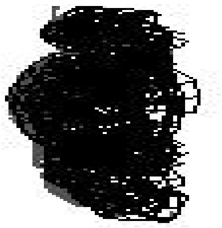


Figure 6.13: View 2 (from above) of model for object “doll” shown overlayed on 3-D data in the final estimated position.

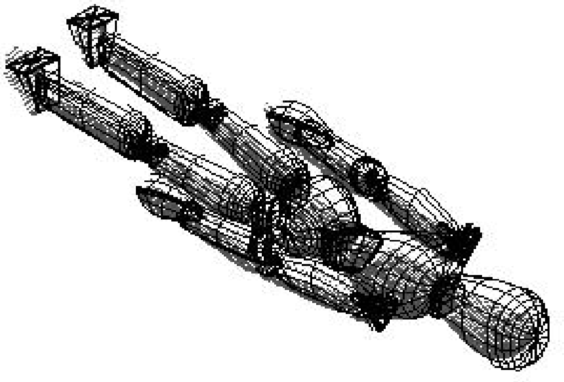


Figure 6.14: View 3 of model for object "doll" shown overlayed on 3-D data in the final estimated position.

Experiments with real data indicate that the pose estimation method described in this chapter produces good results in difficult problems, such as the one approached here. To our knowledge this is the first attempt in 3-D to 3-D pose estimation to deal with parameterized part models represented by superquadrics.

Pose estimation is used in the context of this thesis to locate the hypotheses found in the matching stage of recognition as presented in Chapter 5. The pose estimate is not considered for driving the match at the time the hypotheses are being generated, instead it is mainly for verification of the final hypotheses. Difficulty in estimating the pose of a match may lead to the conclusion that the hypothesis is weak, or it is not fully constrained. By looking at the Figures 6.3 - 6.14, it is possible to evaluate visually the quality of the estimates. This measure is of course subjective, however it provides a reasonable evaluation of the final estimates.

In order to evaluate our approach numerically we have set up simulation experiments for both cases: part-based rigid objects, and part-based non-rigid objects.

Two 3-D objects were generated, both having 3 superquadric components. For the first case, no articulation constraints were imposed on the object. In the second object 2 articulation constraints were specified. A point set with three characteristic points $\mathbf{u}_i = \{\mathbf{gp}_i, \mathbf{map}_i, \mathbf{mip}_i\}$ is determined as the exact positions for each component. Fifty sets of measurements are created by perturbing these exact positions with four different gaussian noise distributions (*i.e.* mean = 0, standard deviation (std) = 0, = 4, = 8, = 16).

These set of points are presented sequentially to the IEKF pose estimation algorithm, and two normalized error measures are computed for each case. The first measure evaluates the error of the estimated translation vector $\hat{\mathbf{t}}$, and it is given by:

$$E_{\hat{\mathbf{t}}} = \frac{1}{3}\sum_{i=1}^{3}\left(\frac{\|\hat{\mathbf{t}}_i - \mathbf{t}_i\|}{\|\mathbf{t}_i\|}\right) \tag{6.46}$$

The second measure evaluates the error of the estimated rotation vector $\hat{\mathbf{s}}$, and it is given by:

$$E_{\hat{\mathbf{s}}} = \frac{1}{3}\sum_{i=1}^{3}\left(\frac{\|\hat{\mathbf{s}}_i - \mathbf{s}_i\|}{\|\mathbf{s}_i\|}\right) \tag{6.47}$$

Figures 6.15 and 6.17 show a graph of error measures for the estimated translation vector $\hat{\mathbf{t}}$, and for the estimated rotation vector $\hat{\mathbf{s}}$, respectively, for the case of a rigid object. Figures 6.16 and 6.18 show the results for the case of a non-rigid object. Curves are shown for four different noise perturbations.

All the graphs show fast convergence and stabilization of the solution. The normalized error for the vector $\hat{\mathbf{t}}$ (Figures 6.15 and 6.16) is reduced to 5% after 5 point sets are fused in both cases, rigid and non-rigid, with noise (std) = 0, and converges for $\leq 3\%$ in every simulated situation. The normalized error for the rotation vector $\hat{\mathbf{s}}$ (Figures 6.17 and 6.18) converges also to a low value ($\leq 6\%$ for the non-rigid case, and $\leq 3\%$ for the rigid case). These results show a good quality of the pose estimated for the type of complex 3-D objects dealt with here.

In the experiments shown only the simple constraint for articulation (Section 6.3.3) was used. This was possible because we restricted the possible movements between parts of the models, and restricted the number of parts that are allowed to rotate (see Appendix D for an example model from the model library). As can be seen from the results, recognition and location of the objects as proposed in this thesis was achieved successfully. However, the methodology for pose estimation including constraints as presented in this chapter can cope with other types of constraints, which could include more elaborate relationships between parts of the objects, such as limiting range and directions of articulations. We believe that future developments in this direction are promising.

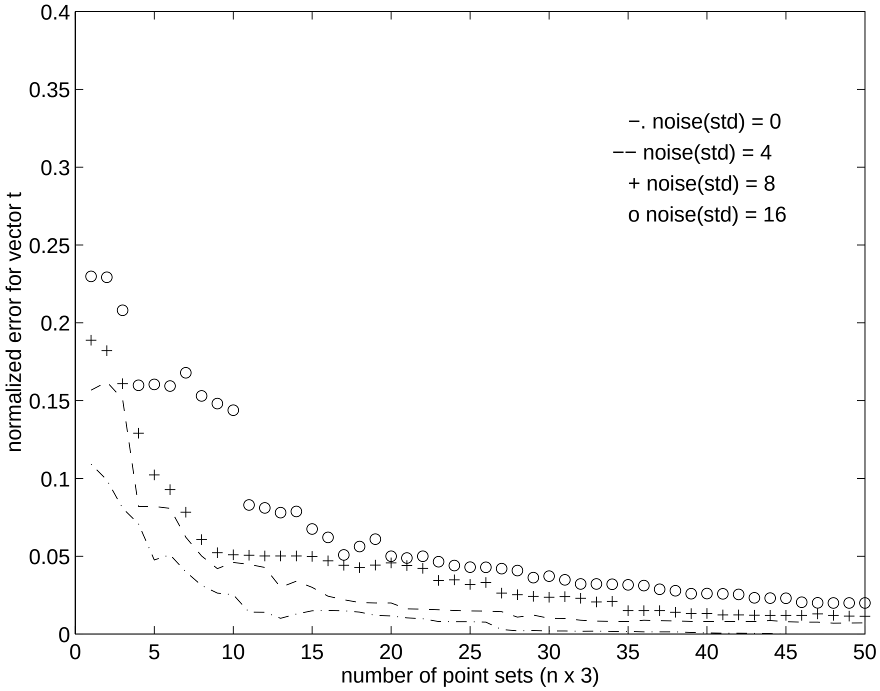


Figure 6.15: Simulation results showing normalized error of estimated translation vector $\hat{\mathbf{t}}$ in function of the number of point sets (n×3) provided, for synthetic rigid object. Four different noise (gaussian) conditions are used to perturb the measurements.

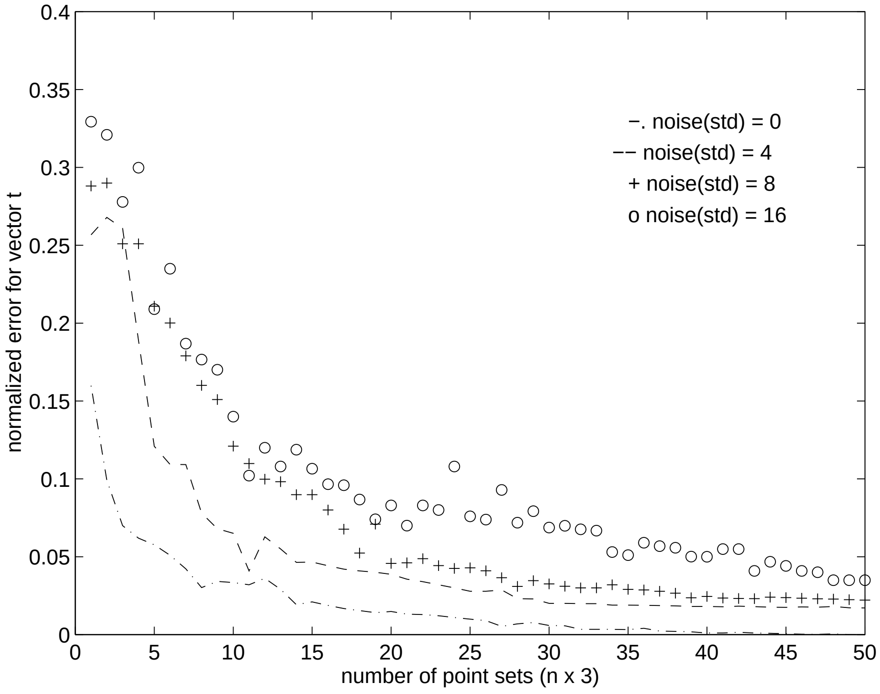


Figure 6.16: Simulation results showing normalized error of estimated translation vector $\hat{\mathbf{t}}$ in function of the number of point sets (n×3) provided, for synthetic non-rigid object. Four different noise (gaussian) conditions are used to perturb the measurements.

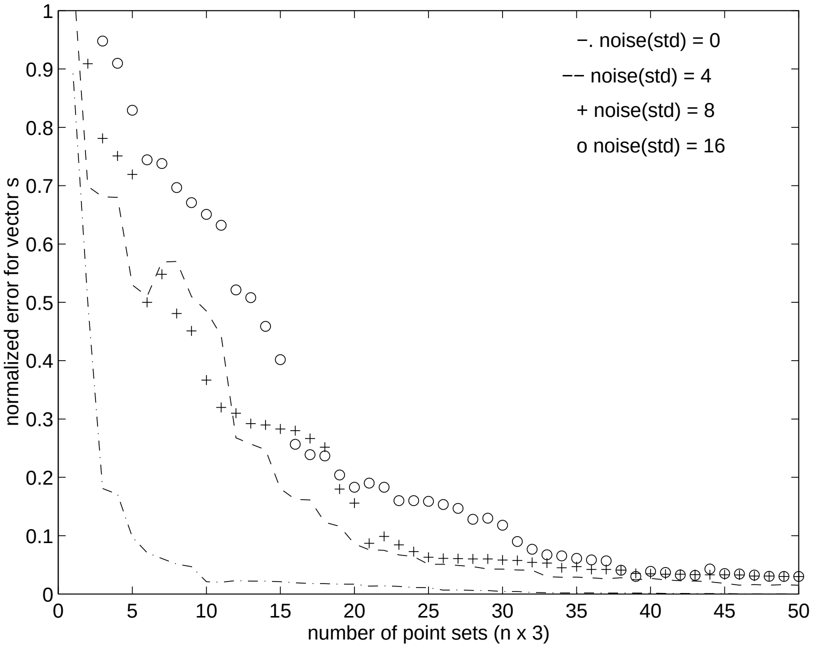


Figure 6.17: Simulation results showing normalized error of estimated rotation vector $\hat{\mathbf{s}}$ in function of the number of point sets (n×3) provided, for synthetic rigid object. Four different noise (gaussian) conditions are used to perturb the measurements.

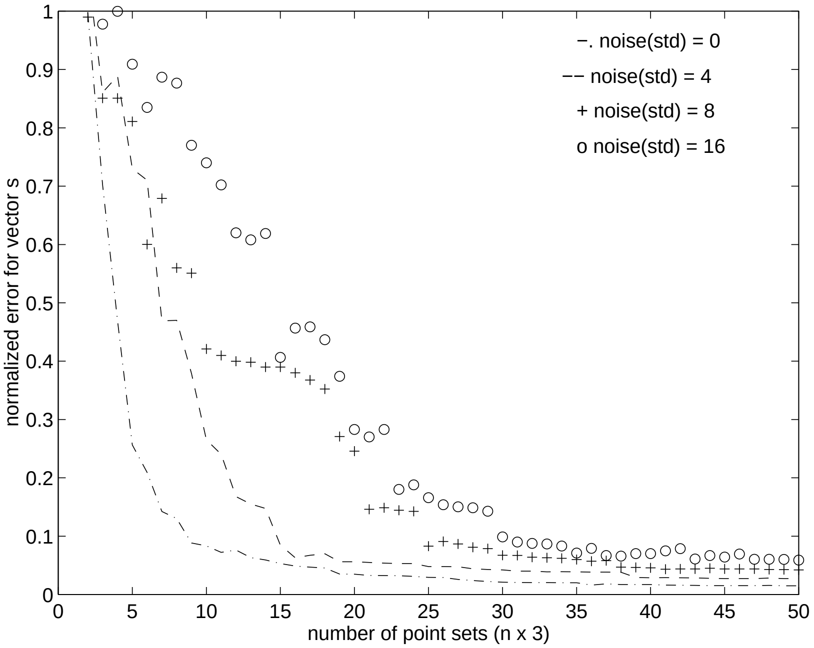


Figure 6.18: Simulation results showing normalized error of estimated rotation vector $\hat{\mathbf{s}}$ in function of the number of point sets (n×3) provided, for synthetic non-rigid object. Four different noise (gaussian) conditions are used to perturb the measurements.

6.5 Discussion

In addition to the set of objects tested shown in this chapter, Appendix F includes results after applying the recognition approach on thirteen (13) more images. Figures F.28, F.29, F.30, F.31, and F.32 show the top hypotheses matched in the final estimated pose, overlayed on the 3-D data. Figures F.33, and F.34 show parts matched for object “pig” overlayed on the 3-D data, however those were not accepted as successful matches.

The pose estimation stage in this thesis is activated after a complete hypothesis is considered acceptable from matching, and as seen from all the tests presented, it performed successfully in every case. Here a successful pose is declared if the estimated position of each of the modelled body parts overlaps the corresponding data region, as assessed visually. The main uncertainty present is a direct consequence of the uncertainty of the superquadric primitives, and the overall performance was satisfactory as can be seen in the overlayed figures.

6.6 Related Work

[Hel-Or 93] developed a general methodology for model-based pose estimation problems based on the Kalman filtering framework. The techniques presented deal with generic point-based models of rigid, articulated, and constrained objects. One of the advantages of its formulation is that it shows a way to include various types of constraints, either linear or non-linear, along with the measurements and their respective uncertainty matrices under an integrated estimation mechanism. The techniques we develop in this thesis for pose estimation of articulated and parameterized objects use a Kalman filtering formulation similar to that presented by Hel-Or in [Hel-Or 93]. In the work presented in this chapter we present an IEKF approach to deal with part-based 3-D volumetric models, represented by superquadrics.

[Kakadiaris *et al.* 94] presented an approach to perform segmentation, shape and motion estimation of articulated objects. The input data consists of a sequence of 2-D images of the objects showing the allowed movements in the articulations. Initially deformable models are fitted into the data, one fitting for each frame of the sequence.

Based on criteria on error of fit, allowable amount of bending deformation and continuity of the data considering the whole sequence, a decision is made to fit two models to that region and identify an articulation. A Kalman filtering technique is then used to determine the joint location.

[Mulligan *et al.* 89] showed a method for arm position sensing of an excavator using 2-D images. Their approach uses quasi-separability, location constraints and edge orientation filtering for determining the joint angles of the arm. The free parameters in the pose are restricted, and computation is performed in one direction only (*i.e.* predefined sequence of constrained movements). By comparing an edge image with a preprocessed one (including parameter estimates and camera models) an error measure is computed. After determining the synthetic image features the error is adjusted iteratively until minimization is achieved.

[Attwood *et al.* 89] presented a technique to recognize human postures using synthetic images. The input is made of synthetic models of a human body composed by straight cylinders. The postures are three: standing, kneeling, and sitting. The technique consists of recovering the major axis of the joints which link the cylinders. The possible rotations are restricted in each direction and the free rotation parameter is found from the homogeneous transformation matrix.

[Grimson 88] addressed the problem of recognizing and locating objects in 2-D images. The objects can vary in parameterized ways, and can differ in scale, stretch, or angles between the parts. Pose estimation is approached by computing a transformation separately for each part of the object, and then assessing its validity by testing it against pre-defined thresholds in an interpretation tree.

The ACRONYM system [Brooks 81] used a constraint manipulation system based on the SUP/INF algorithm to solve for free parameters (symbolic lower and upper bounds) in its representation scheme. One of the proposed parameterizations was articulation, and although results were shown recognizing object classes, articulated objects were not included in the experiments.

[Fisher 89] and later [Fisher & Orr 91] developed a geometric reasoning network based on ACRONYM's constraint manipulation system to cope with parameterizations, and particularly pose parameters and articulations, in 3-D model-based vision systems. Improvement in performance is shown for nonlinear constraints.

A method to parameterize and recognize 3-D polyhedral objects according to scale, stretching and articulations is given by [Reid 92]. Free parameters and pre-computed upper and lower bounds are described in the models. A constraint propagation network based on the SUP/INF method [Fisher & Orr 91] is used for accessing and updating the parameter values. Pose is estimated for each part separately using least squares [Faugeras & Hebert 86], and interpretation is performed by evaluating the range of parameters through the propagation network from a depth-first search of an interpretation tree.

[Lowe 91] described a method to fit parameterized 3-D models to 2-D images. Articulation is included and the method estimates the free parameters which are used to describe the pose by approximating the solution using Newton iterations and Levenberg-Marquadt algorithms for convergence.

[Goldberg 93] presented a rigid data structure which allows for internal parameterization, such as affixment for articulations, to be treated as general viewpoint parameters. This is used further to help compute a correspondence metric and its partial derivatives with respect to pose parameters. A quasi-Newton approximation technique is used for estimating the parameters. Results are shown for a stapler model composed of straight line segments. Only pose estimation considering perfect data is addressed.

[Hogg 83] presented a method which maps sequence of 2-D images depicting a walking person into a description in which the human body is represented by an hierarchical assembly of straight cylinders. Positional, movement, and posture constraints are defined in the model. The moving object is first identified in the sequence by thresholding the images at pixel value, then clustering the thresholded points and enclosing them in minimal rectangles in the image. The walking path is determined by the sequence of centre points of the rectangles. After applying constraints that restrict the movement and positions of the object, linear approximations are carried out to solve for the final pose of the object. Recognition is not addressed, only the tracking of the articulated object, although the methodology could be applied to more general problems.

[Green *et al.* 94] described a functionality based recognition approach dealing with articulated objects. The input of the system is a sequence of plain 3-D objects, but with no part or connection information. The aim is to recognize “generic” objects regarding functionality only, and pose estimation is not addressed.

6.7 Chapter Summary

This chapter presented a method to deal with pose estimation of 3-D constrained volumetric models. The models are represented by assemblies of superquadrics. The method is based around an Iterated Extended Kalman Filter, which is used as the estimation tool where the problem is formulated. The technique is applied as a two stage approach. First, the pose of each part of the model is estimated using characteristic points from the superquadric representation. In the second stage these initial estimates are considered as the *a priori* estimates of an IEKF, the constraints on the model are appended as a single measurement with zero uncertainty, and with the constraint equations linearized they are fused into an IEKF in a batch mode.

The method is applied to estimate the pose of the valid hypotheses found in the matching stage (Chapter 5), and results are shown for four different 3-D complex objects.

The next chapter presents a discussion of the results achieved in all the stages towards recognition considering the main goals set in Chapter 1. Conclusions are drawn in the light of this analysis, and future directions for research are outlined.

Chapter 7

Conclusions

This thesis has developed and demonstrated a method to recognize 3-D complex curved objects using parameterized volumetric models. The approach assumes objects can be represented by simple volumetric parts, and many natural shapes and man-made objects can be modelled by such primitives [Marr & Nishihara 78, Pentland 86]. Recent studies in Psychology suggest that parts structuring plays an important role in object recognition processes [Biederman 87].

The domain of the geometric shapes explored is that of complex curved objects with articulated parts, where there is a great deal of similarity between some of the parts. These objects are exemplified by animal shapes, however the general characteristics and complexity of these shapes are present in a wide range of other natural and man-made 3-D objects.

Most of the previous work in recognition has dealt with objects that can be approximated by a few planar surface patches. Few works have used higher level representation primitives in order to describe complex curved 3-D objects, and to capture the notion of parts structure in depicting the forms. However, those attempts have addressed different, and not all, elements of the model-based 3-D object recognition problem, namely parts-based segmentation and description [Agin 72, Bajcsy & Solina 87, Pentland 90, Gupta 91], and generic symbolic identification without location [Nevatia & Binford 77, Marr & Nishihara 78, Dickinson *et al.* 92b], for a restricted set of forms and with no allowance for articulated parts.

The central issues of recognition of 3-D objects in model-based vision are representation, feature extraction, and interpretation. This thesis has developed and demonstrated an integrated approach that provides answers to each of these three issues in the context of the chosen domain of objects.

To allow for efficiency and flexibility, qualitative representations for shape have to be used along with quantitative ones in order to deal with complex curved objects such as animal shapes. I have developed a composite representation using globally deformable superquadrics to capture quantitative information about the shape; and a set of well-defined volumetric primitives, a subset of geons containing 12 elements, as the qualitative values of the representation. With this composite representation, indexing is more applicable because of discriminative primitives, and the scope of the representation is very large.

Feature extraction comprises a segmentation process whose final goal is to derive a suggestive parts-description of the objects in the scene. This segmentation into parts process exploits a regularity found in 3-D objects which suggests parts division, the transversality regularity [Hoffman & Richards 85]. Three types of discontinuities are extracted from the data, using a controlled diffusion smoothing algorithm and a stable and accurate differential structure of the surface. Depth and concave discontinuities are mapped and preserved before curvature estimation for accuracy. In order to find the closed boundaries of the regions, a dynamic linking of discontinuities is performed by providing each discontinuity point with an active contour, with direction and growth determined by the principal directions of the surface, and heuristics to control length and irrelevant non-closed contours. Contour linking occurs when each isolated region is then fitted by a globally deformable superquadric, which are the basic parts primitives used.

The interpretation issue is addressed in three distinct stages: classification of parts, matching, and pose estimation. A novel solution for classifying superquadric shapes into geons is presented using a Radial Basis Function neural network classifier, where a new training algorithm is developed for the problem. Results were analyzed indicating high performance in comparison with previous work, and high discriminative power for the features which were the objects of study in this work.

Matching in this work was designed to satisfy normal requirements for a search process, *i.e.* efficiency and robustness, and a further requirement of being capable to discriminate between matches of similar objects. These were accomplished by first providing an indexing mechanism, using the qualitative properties of the shape representation and the output of the classifier, to narrow down the search into a set of models which would contain the key volumetric primitives identified in the classification. This indexing mechanism permitted more speed in the search and it can cope with growth of the models library efficiently. The second stage of the matching was designed as a constrained search algorithm, like the Interpretation Tree, using discriminative features from the part primitives and elaborating geometric constraints to drive the search efficiently. Dealing with similar objects, sensitive features, and partial occlusion requires a more sophisticated evaluation of the predicates and valid hypotheses found by the constrained search than threshold limits with yes/no only answers. A set of fuzzy predicates, along with a pre-defined set of weights for quantifying confidence in features and fuzzy sets, was designed to account for these properties. A degree of matching, or a quality measure of similarity, is computed using the framework of Possibility Theory, which provides a way to rank the best valid hypotheses surviving the constrained search. This makes it possible to evaluate similar matches.

Pose estimation, the last component of the interpretation problem, is addressed to provide location of the hypothesis and as a verification step which aids in visualizing the solution of the recognition approach. A way to compute the pose of isolated superquadric volumetric primitives is presented. Simple constraints describing articulations between some of the parts of the objects are included, and then a method to take them into consideration in a unified estimation process is shown. The pose estimation is computed in two stages, first an Iterated Extended Kalman Filter [IEKF] is used to estimate the poses of each of the object's parts, second this vector of transformations is given as the *a priori* measurements to another [IEKF] process with the constraints being given as perfect measurements with no uncertainty. A final pose vector is then estimated and the location of the hypothesis can be visually verified.

Recognition of 3-D objects is a large and difficult problem. This thesis presented an approach to address the issues related to recognition in a model-based paradigm. A domain of geometric objects was chosen, and data were acquired in different conditions for testing and evaluating the approach.

From the four main stages of the approach: segmentation, geon classification, matching, and pose estimation; the most fragile of them is the segmentation. For all the seventeen (17) test scenes, including eleven (11) different objects, one object was not recognized successfully (the object "pig") in four test scenes. This was due mainly to the size and form of the pig's parts (with many small concave patches on its surfaces, like bumps), which made the superquadric estimation converge to bad shapes, and the representation of the object's parts less distinguishable. These characteristics had more impact on the segmentation stage. Suggestions for possible improvements on segmentation are given further in Section 7.2

For the testing performed on this work the stages of geon classification, matching, and pose estimation were in fact reliable and robust. These stages achieved correct results for all the cases when segmentation (stage 1) was satisfactory (final hypotheses shown in Figures 6.3, 6.6, 6.9, 6.12, F.28, F.29 (b), F.30, F.31, F.32), and also for the case when segmentation was unsatisfactory for a portion of the object (object "cheetah", final hypothesis shown in F.29 (a)).

Although the amount of testing done in this work is by no means definitive, the results shown are expected to be representative of many important conditions usually present in this complex domain. A final word would be that this is only a possible approach to the problem of recognition, and many others seem perfectly feasible.

7.1 Summary of Contributions

Major contributions presented in this work include:

- A novel approach to recognizing complex 3-D objects using parameterized parts-based models. The approach addresses the issues of representation, feature extraction, and interpretation providing answers to each of them by exploiting the scenario of objects exemplified by animal shapes.
- A segmentation into parts method that integrates a variety of techniques to provide a solution to parts-based segmentation of 3-D objects. The approach

has three basic stages: first, it computes stable and accurate estimates of the differential structure of the object's surfaces; second, it finds closed boundaries of regions in the data by linking detected surface discontinuities (depth, concave orientation, and negative minima of curvature) using distributed active contours; and third, it fits deformable superquadric models to each of the closed regions.

- A shape representation which features both quantitative and qualitative information is introduced. It has a large scope of applicability, and indexing properties to represent natural articulated 3-D forms.

- A multi-dimensional classifier technique for mapping deformable superquadric shapes into a subset of geon primitives. The classifier is designed using a Radial Basis Function neural network architecture, and an algorithm to train the network to achieve a high generalization score is presented for the problem.

- An indexing technique, that uses the geon classification of the object parts to organize the model library, and to reduce the size of the search space for matching. This is achieved by focusing the first part of the search on models with equivalent indices.

- A constrained search algorithm for matching parameterized parts-models to the parts-descriptions extracted from the range data. The algorithm builds an Interpretation Tree with the ordered hypotheses and features, and decides the pairings by evaluating fuzzy predicates using a pre-defined decision space. This represents precisely ranges of distinctness in the feature values, and attributes different levels of confidence. These local fuzzy measures in the pairings are further used in a global quality measure of the match.

- A similarity metric to evaluate matches. A pre-defined set of weights is given for the feature predicates and for the fuzzy measures. The quality measure of the match describes the degree of membership of the weighted set of variables relative to what would be a perfect match.

- A pose estimation algorithm for articulated objects represented by parameterized volumetric parts. The poses are computed in order to locate and verify the best hypotheses found in the matching stage of recognition. A method for pose estimation of individual deformable superquadric parts is shown, and is used as the initial estimates for computing the pose of articulated objects in a second stage. Both stages are developed in a Kalman Filtering framework. The algorithm shows how to include constraints in the estimation process, and it achieves a final pose integrating all the information.

7.2 Future Directions for Research

The approach developed in this thesis was shown to achieve its primary goal of recognition in the object domain explored. Problems were identified and novel methods were contributed. In this section we list future directions for research uncovered by the results presented here.

- The parts-based segmentation method used only one view of the object from which to derive the parts-description. For symmetric objects, or when that view exposes the biggest portion of the object surface, this segmentation is capable of finding a good description. However, acquiring multiple views of the object and taking into consideration the registered data for parts-segmentation would make the segmentation more robust and the parts-description possibly more accurate, particularly for objects not totally symmetric.

- Other methods can be explored for linking the concave discontinuities found on the object surface. The use of active contours is slow in this case. A composite method that would use both active contours and heuristics to take advantage of inferring boundaries quickly from regions with a high number of discontinuities could be explored.

- The classification of the deformable superquadrics into a subset of 12 geon primitives could be extended to include different sets of volumetric primitives. These primitives could be either direct variations of the basic 12 geons presented, or be defined from extensive fitting and training of the RBF classifier on new objects. Library indexing would have to be extended in the same way incorporating the new primitives.

- More complex constraints defining even more general categories of shapes could be designed and incorporated in the object models. It should be possible to accommodate many different constraints in the estimation method and follow the same framework.

- Within class variation, *e.g.* for an animal shape of a "cow" there are slight differences in the shape of a generic "cow" object that can be represented, perhaps as a hierarchy of models and submodels.

Bibliography

[Agin 72] G.J. Agin. *Representation and Description of Curved Objects.* PhD thesis, Dept. of Computer Science, Stanford University, 1972.

[Arman & Aggarwal 93] F. Arman and J.K. Aggarwal. Model-based object recognition in dense range images - a review. *ACM Computing Surveys*, 25(1):5–43, 1993.

[Attwood *et al.* 89] C.I. Attwood, G.D. Sullivan, and K.D. Baker. Model-based recognition of human posture using single synthetic images. In *Proc. Alvey'89 Conference*, pages 25–30, 1989.

[Bajcsy & Solina 87] R. Bajcsy and F. Solina. Three dimensional object representation revisited. In *Proc. First Int. Conf. Computer Vision, London*, pages 231–240. IEEE Press, 1987.

[Barr 81] A.H. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1(1):11–23, 1981.

[Benson & Evans 77] A. Benson and D.J. Evans. A normalized algorithm for the solution of positive definite symmetric quindiagonal systems of linear equations. *ACM Trans. on Mathematical Software*, 3(1):96–103, 1977.

[Bergevin & Levine 89] R. Bergevin and M. Levine. Building coarse 3D descriptions from line drawings. In *Proc. IEEE Workshop on Interpretation of 3D Scenes, Austin*, pages 68–74. IEEE Press, 1989.

[Bergevin & Levine 92] R. Bergevin and M.D. Levine. Part decomposition of objects from single view drawings. *CVGIP: Image Understanding*, 55(1):73–83, 1992.

[Besl & Jain 85] P.J. Besl and R.C. Jain. Three-dimensional object recognition. *ACM Computing Surveys*, 17(1):75–145, 1985.

[Bhanu 84] B. Bhanu. Representation and shape matching of 3-D objects. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6(3):340–351, 1984.

[Biederman 87] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.

[Binford 82] T.O. Binford. Survey of model-based image analysis systems. *Int. J. Robotics Research*, 1(1):18–64, 1982.

[Bishop 91] C. Bishop. Improving the generalisation properties of radial basis functions neural networks. *Neural Computation*, 3:579–588, 1991.

[Bolles & Horaud 86] R. Bolles and P. Horaud. 3DPO: A three-dimensional part orientation system. *Int. J. Robotics Research*, 5(3):3–26, 1986.

[Borges & Fisher 93] D.L. Borges and R.B. Fisher. Segmentation of 3D articulated objects by dynamic grouping of discontinuities. In J. Illingworth, editor, *Proc. British Machine Vision Conf. 93, Guildford*, pages 279–288. BMVA Press, 1993.

[Borges *et al.* 94] D.L. Borges, M.J. Orr, and R.B. Fisher. A radial basis function neural network for parts identification of three dimensional shapes. In *Proc. VII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI), Curitiba, November 94*. Sociedade Brasileira de Computação, 1994.

[Brady *et al.* 89] J.P. Brady, N. Nandhakumar, and J.K. Aggarwal. Recent progress in object recognition from range data. *Image and Vision Computing*, 7(4):295–307, 1989.

[Brooks 81] R. Brooks. Symbolic reasoning among 3-D models and 2-D images. *Artificial Intelligence*, 17:285–348, 1981.

[Broomhead & Lowe 88] D.S. Broomhead and D. Lowe. Multivariate functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988.

[Buchanan & Shortliffe 84] B.G. Buchanan and E.H. (eds) Shortliffe. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Addison-Wesley, MA, 1984.

[Chen & Kak 89] C.H. Chen and A.C. Kak. A robot vision system for recognizing 3-D objects in low-order polynomial time. *IEEE Trans. Systems, Man, and Cybernetics*, 19(6):1535–1563, 1989.

[Chen *et al.* 91] S. Chen, C.F.N. Cowan, and P.M. Grant. Orthogonal least squares learning for radial basis function networks. *IEEE Trans. Neural Networks*, 2:302–309, 1991.

[Chin & Dyer 86] R.T. Chin and C.R. Dyer. Model-based recognition in robot vision. *ACM Computing Surveys*, 18(1):67–108, 1986.

[Darrell *et al.* 90] T. Darrell, S. Sclaroff, and A. Pentland. Segmentation by minimal description. In *Proc. Third Int. Conf. Computer Vision, Osaka*. IEEE Press, 1990.

[David & Zucker 90] C. David and S.W. Zucker. Potentials, valleys, and dynamic global coverings. *Int. J. Computer Vision*, 5(3):219–238, 1990.

[Dickinson *et al.* 92a] S.J. Dickinson, A. Pentland, and A. Rosenfeld. 3-D shape recovery using distributed aspect matching. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2):174–198, 1992.

[Dickinson *et al.* 92b] S.J. Dickinson, A. Pentland, and A. Rosenfeld. From volumes to views: An approach to 3-D object recognition. *CVGIP: Image Understanding*, 55(2):130–154, 1992.

[Ettinger 87] G.J. Ettinger. Hierarchical object recognition using libraries of parameterized model sub-parts. MSc Thesis, also as Technical Report 963 (AI Lab), Dept. of Electrical Engineering and Computer Science, A.I. Laboratory, M.I.T., 1987.

[Fan 90] T.J. Fan. *Describing and Recognizing 3-D Objects Using Surface Properties.* Springer-Verlag, 1990.

[Faugeras & Hebert 86] O.D. Faugeras and M. Hebert. The representation, recognition, and locating of 3-D objects. *I.J. Robotics Res.*, 5(3):27–52, 1986.

[Ferrie *et al.* 89] F.P. Ferrie, J. Lagarde, and P. Whaite. Darboux frames, snakes, and super-quadrics: Geometry from the bottom-up. Mcgill research centre for intelligent machines, university reports, McGill University, 1989.

[Fisher & Orr 91] R.B. Fisher and M.J.L. Orr. Geometric reasoning in a parallel network. *Int. J. Robotics Research*, 10(2):103–122, 1991.

[Fisher 87] R.B. Fisher. SMS: A suggestive modelling system for object recognition. *Image and Vision Computing*, 5(2):98–194, 1987.

[Fisher 89]	R.B. Fisher. *From Surfaces to Objects: Computer Vision and Three Dimensional Analysis.* John Wiley and Sons, 1989.
[Fisher 94]	R. Fisher. Performance comparison of ten variations on the interpretation-tree matching algorithm. In J.O. Eklundh, editor, *Proc. Third European Conference on Computer Vision 94, Stockholm, Sweden*, volume I, pages 507–512. Springer-Verlag, 1994.
[Flynn & Jain 91]	P.J. Flynn and A.K. Jain. BONSAI: 3-D object recognition using constrained search. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(10):1066–1074, 1991.
[Gardner 65]	M. Gardner. The superellipse: a curve that lies between the ellipse and the rectangle. *Scientific American*, 213:222–236, 1965.
[Geman *et al.* 92]	S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992.
[Goldberg 93]	R.R. Goldberg. Pose determination of parameterized object models from a monocular image. *Image and Vision Computing*, 11(1):49–62, 1993.
[Golub *et al.* 79]	G.H. Golub, M. Heath, and G. Wahba. Generalised cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.
[Green *et al.* 94]	K. Green, D. Eggert, L. Stark, and K. Bowyer. Generic recognition of articulated objects by reasoning about functionality. In *Proc. 12th Int. Conf. Pattern Recognition*, pages 847–849, Israel, 1994.
[Grimson & Lozano-Perez 84]	W.E.L. Grimson and T. Lozano-Perez. Model-based recognition and localization from sparse range or tactile data. *I.J. Robotics Res.*, 3(3):3–35, 1984.
[Grimson 88]	W.E.L. Grimson. On the recognition of parameterized 2D objects. *Int. Journal of Computer Vision*, 2:353–372, 1988.
[Grimson 90]	W.E.L. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints.* MIT Press, 1990.
[Gross & Boult 88]	A.D. Gross and T.E. Boult. Error of fit measures for recovering parametric solids. In *Proc. Second Int. Conf. Computer Vision, Florida.* IEEE Press, 1988.

[Gupta 91] A. Gupta. *Surface and Volumetric Segmentation of Complex 3-D Objects Using Parametric Shape Models.* PhD thesis, Dept. Computer and Information Science, University of Pennsylvania, 1991.

[Hel-Or 93] Y. Hel-Or. *Model based pose estimation from uncertain data.* PhD thesis, Dept. Computer Science, Hebrew University of Jerusalem, 1993.

[Hoffman & Richards 85] D.D. Hoffman and W.A. Richards. Parts of recognition. In S. Pinker, editor, *Visual Cognition*, pages 65–96. MIT Press, 1985.

[Hogg 83] D. Hogg. Model-based vision: A program to see a walking person. *Image and Vision Computing*, 1(1):5–20, 1983.

[Horn 86] B.K.P. Horn. *Robot Vision.* MIT Press, 1986.

[Hughes 91] H.W. Hughes. *Recognition of three dimensional objects using deformable models.* PhD thesis, Dept. Artificial Intelligence, University of Edinburgh, 1991.

[Jackson 89] J.R.H. Jackson. Radial basis functions: a survey and new results. In D.C. Handscomb, editor, *Mathematics of Surfaces III*, pages 115–133. Clarendon Press, Oxford, 1989.

[Jain & Hoffman 88] A.K. Jain and R. Hoffman. Evidence-based recognition of 3-D objects. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 10(6):783–802, 1988.

[Jain & Jain 89] R. Jain and A.K. Jain. Report on range image understanding workshop, east lansing, michigan, march 88. *Machine Vision and Applications*, 2:45–60, 1989.

[Kakadiaris *et al.* 94] I.A. Kakadiaris, D. Metaxas, and R. Bajcsy. Active part-decomposition, shape and motion estimation of articulated objects: A physics-based approach. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 980–984. IEEE Press, 1994.

[Kass *et al.* 87] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int. J. Computer Vision*, 1:321–331, 1987.

[Kim & Kak 91] W.Y. Kim and A.C. Kak. 3-D object recognition using bipartite matching embedded in discrete relaxation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(3):224–251, 1991.

[Leyton 87] M. Leyton. Nested structures of control: An intuitive view. *Computer Vision, Graphics, and Image Processing*, 37:20–53, 1987.

[Lippmann 89] R.P. Lippmann. Pattern classification using neural networks. *IEEE Communications Magazine*, 27:47–64, November 1989.

[Lowe 85] D.G. Lowe. *Perceptual Organization and Visual Recognition.* Kluwer Academic Pub., 1985.

[Lowe 91] D.G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(5):441–450, 1991.

[Luger & Stubblefield 92] G.F. Luger and W.A. Stubblefield. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving.* Benjamin-Cummings, 2nd edition, 1992.

[Marr & Nishihara 78] D. Marr and H.K. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proc. R. Soc. London B*, 200:269–294, 1978.

[Marr 82] D. Marr. *Vision.* W.H. Freeman, 1982.

[Maybeck 79] P.S. Maybeck. *Stochastic Models, Estimation and Control*, volume I,II,III. Academic Press, 1979.

[Mulgaonkar *et al.* 84] P.G. Mulgaonkar, L.G. Shapiro, and R.M. Haralick. Matching 'sticks, plates, and blobs' objects using geometric and relational constraints. *Image and Vision Computing*, 2(2):85–98, 1984.

[Mulligan *et al.* 89] I.J. Mulligan, A.K. Mackworth, and P.D. Lawrence. A model-based vision system for manipulator position sensing. In *Proc. IEEE Workshop on Interpretation of 3D Scenes, Austin*, pages 186–193. IEEE Press, 1989.

[Nevatia & Binford 77] R. Nevatia and T.O. Binford. Description and recognition of curved objects. *Artificial Intelligence*, 8:77–98, 1977.

[Nilsson 80] N.J. Nilsson. *Principles of Artificial Intelligence.* Tioga, CA, 1980.

[Oshima & Shirai 83] M. Oshima and Y. Shirai. Object recognition using three-dimensional information. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 5(4):353–361, 1983.

[Pentland & Sclaroff 91] A. Pentland and S. Sclaroff. Closed-form solutions for physically based shape modeling and recognition. *IEEE Trans. P.A.M.I.*, 13(7):715–729, 1991.

[Pentland 86] A. Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28:293–331, 1986.

[Pentland 87] A Pentland. Recognition by parts. In *Proc. First Int. Conf. Computer Vision, London*, pages 612–620. IEEE Press, 1987.

[Pentland 90] A. Pentland. Automatic extraction of deformable part models. *Int. J. Computer Vision*, 4:107–126, 1990.

[Pollard *et al.* 87] S.B. Pollard, J. Porrill, J.E.W. Mayhew, and J. Frisby. Matching geometrical descriptions in three-space. *Image and Vision Computing*, 5(2):73–78, 1987.

[Powell 87] M.J.D. Powell. Radial basis functions for multivariate approximation: a review. In J.C. Mason and M.G. Cox, editors, *Proc. of IMA Conf. on Algorithms for Approximation of Functions and Data*, pages 143–167. Oxford University Press, 1987.

[Press *et al.* 92] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C.* Cambridge University Press UK), 2nd edition, 1992.

[Raja & Jain 92] N.S. Raja and A.K. Jain. Recognizing geons from superquadrics fitted to range data. *Image and Vision Computing*, 10(3):179–190, 1992.

[Reid & Brady 93] I. Reid and M. Brady. Recognition of object classes from range data. In *Proc. Fourth Int. Conf. Computer Vision, Berlin*, pages 302–307. IEEE Press, 1993.

[Reid 92] I. Reid. *Recognizing parameterized objects from range data.* PhD thesis, Dept. Enginering Science, Oxford University, 1992.

[Roberts 65] L. Roberts. Machine perception of three-dimensional solids. In J.T. Tippett et al., editor, *Optical and Electrooptical Information Processing.* MIT Press, 1965.

[Sander & Zucker 90] P. Sander and S. Zucker. Inferring surface trace and differential structure from 3-D images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(9):833–854, 1990.

[Shapiro.L.G. & Lu 90] Shapiro.L.G. and H. Lu. Accumulator-based inexact matching using relational summaries. *Machine Vision and Applications*, 3:143–158, 1990.

[Solina & Bajcsy 90] F. Solina and R. Bajcsy. Recovery of parametric models from range images: The case for superquadrics with global deformations. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(2):133–147, 1990.

[Solina 87] F. Solina. *Shape Recovery and Segmentation with Deformable Part Models.* PhD thesis, Dept. Computer and Information Science, University of Pennsylvania, 1987.

[Stein & Medioni 92] F. Stein and G. Medioni. Structural indexing: Efficient 3-D object recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2):125–145, 1992.

[Terzopoulos & Metaxas 90] D. Terzopoulos and D. Metaxas. Dynamic 3D models with local and global deformations: Deformable superquadrics. In *Proc. Third Int. Conf. on Computer Vision, Osaka*, pages 606–615. IEEE Press, 1990.

[Terzopoulos 87a] D. Terzopoulos. Matching deformable models to images: Direct and iterative solutions. In *Topical Meeting on Machine Vision, Tec. Digest Series*, volume 12, pages 164–167. SPIE, 1987.

[Terzopoulos 87b] D. Terzopoulos. On matching deformable models to images. In *Topical Meeting on Machine Vision, Tec. Digest Series*, volume 12, pages 160–163. SPIE, 1987.

[Tikhonov & Arsenin 77] A.N. Tikhonov and V.Y. Arsenin. *Solutions of Ill-Posed Problems.* John Wiley, 1977.

[Trucco & Fisher 92] E. Trucco and R. Fisher. Preserving shape at boundaries in diffusion smoothing. In *Proc.IMA Mathematics of Surfaces V, Edinburgh.* Oxford University Press, 1992.

[Trucco & Fisher 94] E. Trucco and R. Fisher. Acquisition of consistent range data using local calibration. In *Proc. IEEE Int. Conf. on Robotics and Automation, San Diego, USA*, volume 4, pages 3410–3415, 1994.

[Zadeh 78] L.A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1(1):3–28, 1978.

[Zucker *et al.* 88] S.W. Zucker, C. David, A. Dobbins, and L. Iverson. The organization of curve detection: Coarse tangent fields and fine spline coverings. In *Proc. Second Int. Conf. Computer Vision, Florida*, pages 568–577. IEEE Press, 1988.

Appendix A

Darboux Frame Calculations

A.1 First Estimates

Consider the surface S parameterized as

$$S = \mathbf{x}(u, v) = (u, v, h(u, v)) \tag{A.1}$$

The following derivatives are obtained from A.1

$$\begin{array}{rlll} \mathbf{x}_u = (1, 0, h_u) & \mathbf{x}_v = (0, 1, h_v) & \\ \mathbf{x}_{uu} = (0, 0, h_{uu}) & \mathbf{x}_{uv} = (0, 0, h_{uv}) & \mathbf{x}_{vv} = (0, 0, h_{vv}) \end{array}$$

The normal of the surface is then given by

$$\mathbf{n} = \frac{\mathbf{x}_u \times \mathbf{x}_v}{\mid \mathbf{x}_u \times \mathbf{x}_v \mid} \quad = \quad \frac{(-h_u, -h_v, 1)}{\sqrt{1 + h_u^2 + h_v^2}} \tag{A.2}$$

Let

$$e = -\langle \mathbf{n}_u, \mathbf{x}_u \rangle = \langle \mathbf{n}, \mathbf{x}_{uu} \rangle = \frac{h_{uu}}{\sqrt{1 + h_u^2 + h_v^2}} \tag{A.3}$$

$$f = -\langle \mathbf{n}_v, \mathbf{x}_u \rangle = -\langle \mathbf{n}_u, \mathbf{x}_v \rangle = \frac{h_{uv}}{\sqrt{1 + h_u^2 + h_v^2}} \tag{A.4}$$

$$g = -\langle \mathbf{n}_v, \mathbf{x}_v \rangle = \langle \mathbf{n}, \mathbf{x}_{vv} \rangle = \frac{h_{vv}}{\sqrt{1 + h_u^2 + h_v^2}} \tag{A.5}$$

and

$$E = \langle \mathbf{x}_u, \mathbf{x}_u \rangle = 1 + h_u^2 \tag{A.6}$$

$$F = \langle \mathbf{x}_u, \mathbf{x}_v \rangle = \langle \mathbf{x}_v, \mathbf{x}_u \rangle = h_u h_v = h_v h_u \tag{A.7}$$

$$G = \langle \mathbf{x}_v, \mathbf{x}_v \rangle = 1 + h_v^2 \tag{A.8}$$

where $\langle , \rangle$ stands for the inner product between the two vectors.

And, calculating the *Weingarten equations*,

$$
\begin{aligned}
d\mathbf{n}\begin{pmatrix} du \\ dv \end{pmatrix} &= \begin{pmatrix} \frac{fF-eG}{EG-F^2} & \frac{gF-fG}{EG-F^2} \\ \\ \frac{eF-fE}{EG-F^2} & \frac{fF-gE}{EG-F^2} \end{pmatrix}\begin{pmatrix} du \\ dv \end{pmatrix} \\
d\mathbf{n}\begin{pmatrix} du \\ dv \end{pmatrix} &= \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}\begin{pmatrix} du \\ dv \end{pmatrix} \\
d\mathbf{n}\begin{pmatrix} du \\ dv \end{pmatrix} &= \mathbf{A}\begin{pmatrix} du \\ dv \end{pmatrix} \qquad \text{(A.9)}
\end{aligned}
$$

The Gaussian curvature K is obtained by

$$\mathrm{K} = det\mathbf{A} \qquad \text{(A.10)}$$

and the Mean curvature H from

$$\mathrm{H} = -\frac{1}{2}(a_{11} + a_{22}) \qquad \text{(A.11)}$$

The principal curvatures (κ_1, κ_2) are the eigenvalues of $\mathbf{A}$,

$$
\begin{aligned}
-\kappa_1 &= \frac{1}{2}(a_{11} + a_{22} - \sqrt{(a_{11} - a_{22})^2 + 4a_{12}a_{21}}) \qquad \text{(A.12)} \\
-\kappa_2 &= \frac{1}{2}(a_{11} + a_{22} + \sqrt{(a_{11} - a_{22})^2 + 4a_{12}a_{21}}) \qquad \text{(A.13)} \\
\kappa_1 \geq \kappa_2 &\quad and \quad (a_{11} - a_{22})^2 + 4a_{12}a_{21} \geq 0
\end{aligned}
$$

and the eigenvectors of $\mathbf{A}$ are the principal directions $\psi_{1,2}$

$$
\begin{aligned}
\psi_1 &= \left(a_{12}, -\frac{1}{2}(a_{11} - a_{22} + \sqrt{(a_{11} - a_{22})^2 + 4a_{12}a_{21}})\right) \quad a_{11} \geq a_{22} \\
&\quad \left(\frac{1}{2}(a_{11} - a_{22} - \sqrt{(a_{11} - a_{22})^2 + 4a_{12}a_{21}}), a_{21}\right) \quad a_{11} < a_{22} \qquad \text{(A.14)} \\
\psi_2 &= \left(\frac{1}{2}(a_{11} - a_{22} + \sqrt{(a_{11} - a_{22})^2 + 4a_{12}a_{21}}), a_{21}\right) \quad a_{11} \geq a_{22} \\
&\quad \left(-a_{12}, -\frac{1}{2}(a_{11} - a_{22} - \sqrt{(a_{11} - a_{22})^2 + 4a_{12}a_{21}})\right) \quad a_{11} < a_{22} \qquad \text{(A.15)}
\end{aligned}
$$

A.2 Enforcing Consistency

To estimate Γ_P from the support neighbourhood Γ_{Q_α} the values are extrapolated from a paraboloid S_{Q_α}, as:

$$S_{Q_\alpha} = (u, v, \frac{1}{2}(\kappa_{1(\alpha)}u^2 + \kappa_{2(\alpha)}v^2)) \qquad \text{(A.16)}$$

and the local tangent plane basis vectors are:

$$
\begin{aligned}
P(u,v) &= \tfrac{\partial S}{\partial u}(u,v) = (1, 0, \kappa_{1(\alpha)}u) \qquad \text{(A.17)} \\
Q(u,v) &= \tfrac{\partial S}{\partial v}(u,v) = (0, 1, \kappa_{2(\alpha)}v) \qquad \text{(A.18)}
\end{aligned}
$$

In this basis the coefficients for the first and second fundamental forms are:

$$E(u,v) = \langle S_u, S_u \rangle = \kappa_{1(\alpha)}^2 u^2 + 1 \tag{A.19}$$

$$F(u,v) = \langle S_u, S_v \rangle = \kappa_{1(\alpha)} \kappa_{2(\alpha)} uv \tag{A.20}$$

$$G(u,v) = \langle S_v, S_v \rangle = \kappa_{2(\alpha)}^2 v^2 + 1 \tag{A.21}$$

$$e(u,v) = \langle \mathbf{n}_g, S_{uu} \rangle = \frac{\kappa_{1(\alpha)}}{\sqrt{(\kappa_{1(\alpha)}^2 u^2 + \kappa_{2(\alpha)}^2 v^2 + 1)}} \tag{A.22}$$

$$f(u,v) = \langle \mathbf{n}_g, S_{uv} \rangle = 0 \tag{A.23}$$

$$g(u,v) = \langle \mathbf{n}_g, S_{vv} \rangle = \frac{\kappa_{2(\alpha)}}{\sqrt{(\kappa_{1(\alpha)}^2 u^2 + \kappa_{2(\alpha)}^2 v^2 + 1)}} \tag{A.24}$$

and,

$$\mathbf{n}_g(u,v) = \frac{1}{\sqrt{(\kappa_{1(\alpha)}^2 u^2 + \kappa_{2(\alpha)}^2 v^2 + 1)}} (-\kappa_{1(\alpha)} u, -\kappa_{2(\alpha)} v, 1) \tag{A.25}$$

is the surface normal at P. The differential of the surface normal at (u,v) is given by:

$$d\mathbf{n} \begin{pmatrix} du \\ dv \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} du \\ dv \end{pmatrix}$$

with,

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} \frac{fF-eG}{EG-F^2} & \frac{gF-fG}{EG-F^2} \\ \frac{eF-fE}{EG-F^2} & \frac{fF-gE}{EG-F^2} \end{pmatrix}$$

and the principal curvatures, and principal directions are computed as in (A.12), (A.13), (A.14), (A.15).

Appendix B

The Snake Model

The snake model can be described in terms of a Lagrangian formulation of motion [Terzopoulos 87b].

Consider a deformable curve $v(s,t)$, where s is a spatial index and t is time, on given open intervals Ω and T respectively. We define this curve as a function of the coordinate variables x and y as:

$$v(s,t) \quad = \quad (x(s,t), y(s,t)) \qquad : s \in \Omega, t \in T. \tag{B.1}$$

The potential energy function of the snake is defined as

$$E_{snake} \quad = \quad \frac{1}{2} \int_{\Omega} \left[E_{int}(v) + E_{ext}(v) + E_{field}(v) \right] ds \tag{B.2}$$

E_{int} represents the internal energy,

$$E_{int}\left(v(s)\right) \quad = \quad \omega_1(s) \mid v_s \mid^2 + \omega_2(s) \mid v_{ss} \mid^2 \tag{B.3}$$

where,

$$v_s \equiv \frac{\partial v}{\partial s} \quad v_{ss} \equiv \frac{\partial^2 v}{\partial s^2}$$

The external energy E_{ext} arises from basically two types of forces, pushing and pulling [Kass *et al.* 87], modelled as springs and volcanos. Springs are used to attach the snake to special points, and volcanos are used to push the snake out of local minima of the potential field.

The potential field energy is derived from classical gravitational potential energy as [Terzopoulos 87b]:

$$E_{field}(v(s,t)) \quad = \quad \mu \rho z(v(s,t)) \tag{B.4}$$

where,

μ is the constant mass density of the snake.
ρ is the constant magnitude of the field.
$z(v(s,t))$ is the height of snake pixel at time t on the surface.

For a given initial position a minimization is applied to B.2 for the system to reach a stable energy state by converting potential energy to kinetic energy, and dissipating this by an energy dissipation function. For this case the Euler-Lagrange equations of motion apply,

$$\begin{aligned}\mu x_{tt} + \gamma x_t - \frac{\partial}{\partial s}\left(\omega_1(s) x_s\right) + \frac{\partial^2}{\partial s^2}\left(\omega_2(s) x_{ss}\right) &= -\frac{1}{2}\left(E_{extx}(v) + E_{field_x}(v)\right) \\ \mu y_{tt} + \gamma y_t - \frac{\partial}{\partial s}\left(\omega_1(s) y_s\right) + \frac{\partial^2}{\partial s^2}\left(\omega_2(s) y_{ss}\right) &= -\frac{1}{2}\left(E_{exty}(v) + E_{field_y}(v)\right) \quad \text{(B.5)}\end{aligned}$$

where,

$$\begin{aligned} x_{tt} &\equiv \frac{\partial^2 x}{\partial t^2} & y_{tt} &\equiv \frac{\partial^2 y}{\partial t^2} \\ E_{extx} &\equiv \frac{\partial}{\partial x}(E_{ext}) & E_{exty} &\equiv \frac{\partial}{\partial y}(E_{ext}) \\ E_{field_x} &\equiv \frac{\partial}{\partial x}(E_{field}) & E_{field_y} &\equiv \frac{\partial}{\partial y}(E_{field}) \end{aligned}$$

and γ is the viscosity factor.

B.1 Numerical solution for the snake model

The system above B.5 relates the positions, both in space and time, of the curve v in motion. Solution to B.5 can be numerically approximated as the following [Terzopoulos 87a]:

The space domain Ω is tesselated into $N+1$ nodes $\{0, h, 2h, ..., Nh\}$, with $h = 1/N$. The solutions to the system will then consist of vectors

$$\mathbf{x}^t = (x(ih,t))_{i=0}^{N} \qquad \mathbf{y}^t = (y(ih,t))_{i=0}^{N} \tag{B.6}$$

at each iteration time t.

Equation B.5 can be expressed in matrix form as

$$\mathbf{A}\mathbf{x}^t = g_x^t \qquad \mathbf{A}\mathbf{y}^t = g_y^t \tag{B.7}$$

Assuming initially that the curve is closed, two prior configurations are needed. Given a time step Δt, solving for $\mathbf{x}^t$, x_t and x_{tt} are approximated using backward differences by

$$x_t = \frac{x^t - x^{t-2\Delta t}}{2\Delta t} \tag{B.8}$$

$$x_{tt} = \frac{x^{t-2\Delta t} - 2x^{t-\Delta t} + x^t}{\Delta t^2} \tag{B.9}$$

Replacing it and solving for g_x^t,

$$\begin{aligned} g_x^t &= [-\frac{1}{2}(E_{extx}(x_i^{t-\Delta t}, y_i^{t-\Delta t}) + E_{field_x}(x_i^{t-\Delta t}, y_i^{t-\Delta t})) + \\ & \quad 2\frac{\mu}{\Delta t^2}x_i^{t-\Delta t} - \left(\frac{\mu}{\Delta t^2} - \frac{\gamma}{2\Delta t}\right) x_i^{t-2\Delta t}]_{i=0}^N \end{aligned} \tag{B.10}$$

and writing

$$\mathbf{A} = \left[\frac{\mu}{\Delta t^2} + \frac{\gamma}{2\Delta t}\right] \mathbf{I} + \mathbf{K} = \alpha \mathbf{I} + \mathbf{K} \tag{B.11}$$

yields

$$(\alpha \mathbf{I} + \mathbf{K})\mathbf{x}^t = g_x^t \tag{B.12}$$

where the matrix **K**, called the stiffness matrix, determines the relations between the nodes.

Approximating the spatial derivatives by finite differences we get

$$\begin{aligned} -\frac{\partial}{\partial s}\omega_2(x_s) + \frac{\partial^2}{\partial s^2}\omega_2(x_{ss}) \approx\; & x_{i-2h}\left[\frac{\omega_2(i-h)}{h^4}\right] + \\ & x_{i-h}\left[-\frac{2\omega_2(i)}{h^4} - \frac{2\omega_2(i-h)}{h^4} - \frac{\omega_1(i-h)}{h^2}\right] + \\ & x_i\left[\frac{\omega_2(i+h)}{h^4} + \frac{4\omega_2(i)}{h^4} + \frac{\omega_2(i-h)}{h^4} + \frac{\omega_1(i)}{2h^2} - \frac{\omega_1(i-h)}{h^2}\right] + \\ & x_{i+h}\left[-\frac{2\omega_2(i+h)}{h^4} - \frac{2\omega_2(i)}{h^4} - \frac{\omega_1(i)}{h^2}\right] + \\ & x_{i+2h}\left[\frac{\omega_2(i+h)}{h^4}\right] \end{aligned} \tag{B.13}$$

where for closed curves,

$$x_{-1}^t = x_N^t, \quad x_{-2}^t = x_{N-1}^t, \quad x_{N+1}^t = x_0^t, \quad x_{N+2}^t = x_1^t$$

Rearranging formula B.13 by choosing

$$\begin{aligned} a_i &= \frac{\omega_2(i+h)}{h^4} \\ b_i &= -\frac{2\omega_2(i)}{h^4} - \frac{2\omega_2(i+h)}{h^4} - \frac{\omega_1(i)}{h^2} \\ c_i &= \frac{\omega_2(i-h)}{h^4} + \frac{4\omega_2(i)}{h^4} + \frac{\omega_2(i+h)}{h^4} + \frac{\omega_1(i-h)}{h^2} + \frac{\omega_1(i)}{h^2} \end{aligned}$$

The stiffness matrix $\mathbf{K}$ is a pentadiagonal matrix as shown in B.14

$$\mathbf{K} = \begin{pmatrix} c_0 & b_0 & a_0 & & & & a_{N-1} & b_N \\ b_0 & c_1 & b_1 & a_1 & & & & a_N \\ a_0 & b_1 & c_2 & b_2 & a_2 & & & \\ & a_1 & b_2 & c_3 & b_3 & a3 & & \\ & & & \cdot & & & & \\ & & & & \cdot & & & \\ & & & & & \cdot & & \\ & & & a_{N-4} & b_{N-3} & c_{N-2} & b_{N-2} & a_{N-2} \\ a_{N-1} & & & & a_{N-3} & b_{N-2} & c_{N-1} & b_{N-1} \\ b_N & a_N & & & & a_{N-2} & b_{N-1} & c_N \end{pmatrix} \qquad \text{(B.14)}$$

$\mathbf{A} = \alpha\mathbf{I} + \mathbf{K}$ is also a pentadiagonal matrix and can be solved by a known algorithm (O(N) time) such as the one by [Benson & Evans 77].

B.2 Inserting discontinuities

Following [Terzopoulos 87b] in order to insert a position discontinuity between node $N+1$ and node 0, the stiffness matrix $\mathbf{K}$ is to be changed by setting

$$b_{i-1} = a_{i-2} = a_{i-1} = 0$$

which makes

$$\omega_2(i-1) = \omega_2(i) = \omega_1(i-1) = 0$$

The following nine (9) entries are to be changed in $\mathbf{K}$,

$$\begin{aligned} a_{i-2} &= 0 \\ a_{i-1} &= 0 \\ b_{i-2} &= -\frac{2\omega_2(i-2h)}{h^4} - \frac{\omega_1(i-2h)}{h^2} \\ b_{i-1} &= 0 \\ b_i &= -\frac{2\omega_2(i+h)}{h^4} - \frac{\omega_1(i)}{h^2} \\ c_{i-2} &= \frac{\omega_2(i-3h)}{h^4} + \frac{\omega_2(i-2h)}{h^4} \\ c_{i-1} &= \frac{\omega_2(i-2h)}{h^4} + \frac{\omega_1(i-2h)}{h^2} \\ c_i &= \frac{\omega_2(i+h)}{h^4} + \frac{\omega_1(i)}{h^2} \\ c_{i+1} &= \frac{4\omega_2(i+h)}{h^4} + \frac{\omega_2(i+2h)}{h^4} + \frac{\omega_1(i)}{h^2} + \frac{\omega_1(i+h)}{h^2} \end{aligned}$$

Appendix C

Superquadrics Deformations: Tapering and Bending

The definitions for superquadric deformations, tapering and bending, follow [Solina & Bajcsy 90], and are given here for completeness.

Global Tapering Deformation
Tapering deformation is defined along axis z, as:

$$\begin{aligned} X &= f_x(z)x \\ Y &= f_y(z)y \\ Z &= z \end{aligned} \tag{C.1}$$

where, X, Y, Z represent the components of the surface vector $\mathbf{x}$ (deformed); f_x, f_y are the tapering functions; and x, y, z represent the original surface vector $\mathbf{x}$.

For linear tapering, as used in this thesis,

$$f_x(Z_s = \frac{t_x}{a_3} Z_s + 1 \tag{C.2}$$

$$f_y(Z_s = \frac{t_y}{a_3} Z_s + 1 \tag{C.3}$$

where, $(-1 \leq t_x, t_y \leq 1)$.

Bending Deformation
The bending plane is defined as axis z and vector $\mathbf{r}$ in plane $\mathbf{x}$-$\mathbf{y}$, in direction α.

$$r = \cos(\alpha - \beta)\sqrt{(}x^2 + y^2) \tag{C.4}$$

where,

$$\beta = tan^{-1}(y/x) \tag{C.5}$$

Bending transforms r into

$$R = k^{-1} - \cos\gamma(k^{-1} - r) \tag{C.6}$$

and,

$$\gamma \;=\; zk^{-1} \tag{C.7}$$

where k represents the curvature. Finally, the coordinates to be included in the estimation are related as:

$$\begin{aligned} x &= X - \cos\alpha(R-r) \\ y &= Y - \sin\alpha(R-r) \\ z &= k^{-1}\gamma \end{aligned} \tag{C.8}$$

Figure C.1 shows a graph with the definitions in the x-y plane and the bending plane.

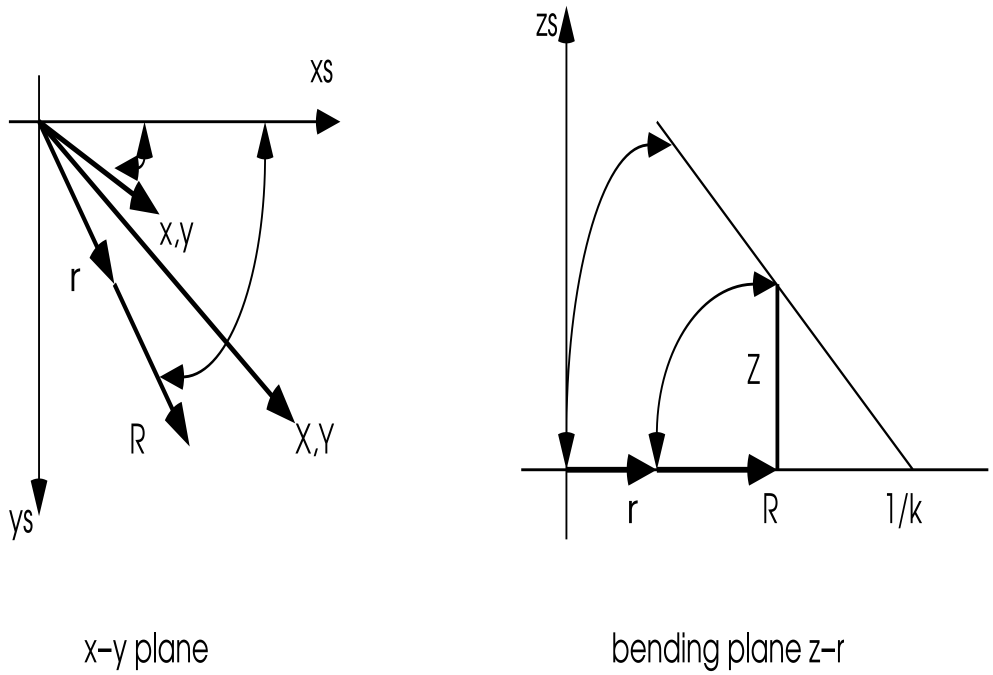


Figure C.1: Definitions for the bending deformation.

Appendix D

An Example Model from the Database

This appendix shows a complete model for the "doll". The identifiers for the data structures used have the following meaning:

- SUP_FEATURE - geometric primitive type (superquadric);
- NAME - label for the part;
- GEON_TYPE - label for one of the twelve geon types;
- POSITION TRANS - x, y, z position;
- POSITIONS RST - RST angles of rotation;
- AX_SUP - size of axis in x of the superquadric;
- AY_SUP - size of axis in y of the superquadric;
- AZ_SUP - size of axis in z of the superquadric;
- EPSILON_1 - ϵ_1 parameter of the superquadric;
- EPSILON_2 - ϵ_2 parameter of the superquadric;
- TAPER_X_DEFORM - tapering deformation in x;
- TAPER_Y_DEFORM - tapering deformation in y;
- RADIUS_BENDING - radius of bending deformation;
- ADJACENCY - parts' adjacencies;
- ARTICULATION NAME - label of the articulated part;
- ARTICULATION DIST - distance limit allowed for articulation;

Model *mod1_doll* from the database:

```
SUP_FEATURE : SUPERQUADRIC
NAME : head
GEON_TYPE : S_C_TA
POSITION
TRANS: (48.0 101.0 18.5)
ROTATION
RST: (3.0  -1.5  -4.7)
AX_SUP: 15.0
AY_SUP: 22.0
AZ_SUP: 32.0
EPSILON_1: 0.2
EPSILON_2: 0.5
TAPER_X_DEFORM: 0.6
TAPER_Y_DEFORM: 0.6
RADIUS_BENDING: 20.0

SUP_FEATURE : SUPERQUADRIC
NAME : neck
GEON_TYPE : S_C_CO
POSITION
TRANS: (87.0 66.5 21.5)
ROTATION
RST: (4.9 -1.3 -0.2)
AX_SUP: 8.0
AY_SUP: 9.5
AZ_SUP: 11.0
EPSILON_1: 0.1
EPSILON_2: 0.5
TAPER_X_DEFORM: 0.05
TAPER_Y_DEFORM: 0.05
RADIUS_BENDING: 15.0

SUP_FEATURE : SUPERQUADRIC
NAME : chest
GEON_TYPE : S_C_TA
POSITION
TRANS: (116.0 98.5 22.0)
ROTATION
RST: (3.0 -1.4 1.6)
AX_SUP: 14.0
AY_SUP: 18.5
AZ_SUP: 40.0
EPSILON_1: 0.2
EPSILON_2: 0.5
TAPER_X_DEFORM: 0.6
TAPER_Y_DEFORM: 0.6
RADIUS_BENDING: 20.0
```

```
SUP_FEATURE : SUPERQUADRIC
NAME : belly
GEON_TYPE : S_C_CO
POSITION
TRANS: (133.0 95.0 21.0)
ROTATION
RST: (4.8 -0.9 -0.2)
AX_SUP: 8.5
AY_SUP: 13.0
AZ_SUP: 16.0
EPSILON_1: 0.1
EPSILON_2: 0.5
TAPER_X_DEFORM: 0.05
TAPER_Y_DEFORM: 0.05
RADIUS_BENDING: 15.0


SUP_FEATURE : SUPERQUADRIC
NAME : shoulder_r
GEON_TYPE : S_C_CO
POSITION
TRANS: (178.5 104.5 20.0)
ROTATION
RST: (4.6 -1.5 0.0)
AX_SUP: 4.0
AY_SUP: 7.0
AZ_SUP: 9.5
EPSILON_1: 0.1
EPSILON_2: 0.5
TAPER_X_DEFORM: 0.05
TAPER_Y_DEFORM: 0.05
RADIUS_BENDING: 15.0


SUP_FEATURE : SUPERQUADRIC
NAME : shoulder_l
GEON_TYPE : S_C_CO
POSITION
TRANS: (94.0 130.0 20.0)
ROTATION
RST: (4.9 -1.4 -0.2)
AX_SUP: 4.0
AY_SUP: 7.0
AZ_SUP: 9.5
EPSILON_1: 0.1
EPSILON_2: 0.5
TAPER_X_DEFORM: 0.05
```

```
TAPER_Y_DEFORM: 0.05
RADIUS_BENDING: 15.0


SUP_FEATURE : SUPERQUADRIC
NAME : upper_arm_r
GEON_TYPE : S_C_TA
POSITION
TRANS: (120.0 132.0 14.0)
ROTATION
RST: (3.2 -1.6 3.0)
AX_SUP: 8.0
AY_SUP: 13.0
AZ_SUP: 24.0
EPSILON_1: 0.1
EPSILON_2: 0.5
TAPER_X_DEFORM: 0.5
TAPER_Y_DEFORM: 0.5
RADIUS_BENDING: 20.0


SUP_FEATURE : SUPERQUADRIC
NAME : upper_arm_l
GEON_TYPE : S_C_TA
POSITION
TRANS: (117.0 63.0 14.0)
ROTATION
RST: (3.0 -1.5 2.8)
AX_SUP: 8.0
AY_SUP: 13.0
AZ_SUP: 24.0
EPSILON_1: 0.1
EPSILON_2: 0.5
TAPER_X_DEFORM: 0.5
TAPER_Y_DEFORM: 0.5
RADIUS_BENDING: 20.0

SUP_FEATURE : SUPERQUADRIC
NAME : elbow_r
GEON_TYPE : S_C_CO
POSITION
TRANS: (178.0 134.0 14.0)
ROTATION
RST: (3.2 -1.5 1.2)
AX_SUP: 3.5
AY_SUP: 5.0
AZ_SUP: 5.0
EPSILON_1: 0.1
```

```
EPSILON_2: 0.5
TAPER_X_DEFORM: 0.05
TAPER_Y_DEFORM: 0.05
RADIUS_BENDING: 15.0

SUP_FEATURE : SUPERQUADRIC
NAME : elbow_l
GEON_TYPE : S_C_CO
POSITION
TRANS: (148.5 60.0 21.0)
ROTATION
RST: (3.2 -1.5 1.2)
AX_SUP: 3.5
AY_SUP: 5.0
AZ_SUP: 5.0
EPSILON_1: 0.1
EPSILON_2: 0.5
TAPER_X_DEFORM: 0.05
TAPER_Y_DEFORM: 0.05
RADIUS_BENDING: 15.0

SUP_FEATURE : SUPERQUADRIC
NAME : lower_arm_r
GEON_TYPE : S_C_TA
POSITION
TRANS: (176.0 134.0 14.0)
ROTATION
RST: (3.0 -1.5 1.2)
AX_SUP: 9.0
AY_SUP: 18.0
AZ_SUP: 30.0
EPSILON_1: 0.1
EPSILON_2: 0.5
TAPER_X_DEFORM: 0.5
TAPER_Y_DEFORM: 0.5
RADIUS_BENDING: 20.0

SUP_FEATURE : SUPERQUADRIC
NAME : lower_arm_l
GEON_TYPE : S_C_TA
POSITION
TRANS: (175.0 55.0 14.0)
ROTATION
RST: (3.0 -1.5 1.2)
AX_SUP: 9.0
AY_SUP: 18.0
AZ_SUP: 30.0
EPSILON_1: 0.1
```

```
EPSILON_2: 0.5
TAPER_X_DEFORM: 0.5
TAPER_Y_DEFORM: 0.5
RADIUS_BENDING: 20.0

SUP_FEATURE : SUPERQUADRIC
NAME : hand_r
GEON_TYPE : S_C_TA
POSITION
TRANS: (205.0 140.0 10.5)
ROTATION
RST: (2.9 -1.6 -0.1)
AX_SUP: 6.0
AY_SUP: 11.0
AZ_SUP: 18.0
EPSILON_1: 0.2
EPSILON_2: 0.6
TAPER_X_DEFORM: 0.7
TAPER_Y_DEFORM: 0.6
RADIUS_BENDING: 12.0

SUP_FEATURE : SUPERQUADRIC
NAME : hand_l
GEON_TYPE : S_C_TA
POSITION
TRANS: (205.0 50.0 10.0)
ROTATION
RST: (2.9 -1.6 -0.1)
AX_SUP: 6.0
AY_SUP: 11.0
AZ_SUP: 18.0
EPSILON_1: 0.2
EPSILON_2: 0.6
TAPER_X_DEFORM: 0.7
TAPER_Y_DEFORM: 0.6
RADIUS_BENDING: 12.0

SUP_FEATURE : SUPERQUADRIC
NAME : hip
GEON_TYPE : S_C_TA
POSITION
TRANS: (155.0 95.5 17.5)
ROTATION
RST: (4.8 -1.5 -3.7)
AX_SUP: 19.0
AY_SUP: 20.0
AZ_SUP: 22.0
EPSILON_1: 0.1
```

```
EPSILON_2: 0.5
TAPER_X_DEFORM: 0.5
TAPER_Y_DEFORM: 0.5
RADIUS_BENDING: 20.0

SUP_FEATURE : SUPERQUADRIC
NAME : upper_thigh_r
GEON_TYPE : S_C_CO
POSITION
TRANS: (72.0 100.0 21.0)
ROTATION
RST: (4.6 -1.4 0.0)
AX_SUP: 3.5
AY_SUP: 7.0
AZ_SUP: 8.0
EPSILON_1: 0.1
EPSILON_2: 0.5
TAPER_X_DEFORM: 0.05
TAPER_Y_DEFORM: 0.05
RADIUS_BENDING: 15.0

SUP_FEATURE : SUPERQUADRIC
NAME : upper_thigh_l
GEON_TYPE : S_C_CO
POSITION
TRANS: (178.0 90.0 21.0)
ROTATION
RST: (4.6 -1.4 0.0)
AX_SUP: 3.5
AY_SUP: 7.0
AZ_SUP: 8.0
EPSILON_1: 0.1
EPSILON_2: 0.5
TAPER_X_DEFORM: 0.05
TAPER_Y_DEFORM: 0.05
RADIUS_BENDING: 15.0

SUP_FEATURE : SUPERQUADRIC
NAME : thigh_r
GEON_TYPE : S_C_TA
POSITION
TRANS: (216.0 112.0 13.5)
ROTATION
RST: (3.1 -1.7 1.2)
AX_SUP: 9.0
AY_SUP: 16.0
AZ_SUP: 35.0
EPSILON_1: 0.1
```

```
EPSILON_2: 0.5
TAPER_X_DEFORM: 0.5
TAPER_Y_DEFORM: 0.5
RADIUS_BENDING: 20.0

SUP_FEATURE : SUPERQUADRIC
NAME : thigh_l
GEON_TYPE : S_C_TA
POSITION
TRANS: (216.0 77.0 13.5)
ROTATION
RST: (3.1 -1.7 1.2)
AX_SUP: 9.0
AY_SUP: 16.0
AZ_SUP: 35.0
EPSILON_1: 0.1
EPSILON_2: 0.5
TAPER_X_DEFORM: 0.5
TAPER_Y_DEFORM: 0.5
RADIUS_BENDING: 20.0

SUP_FEATURE : SUPERQUADRIC
NAME : knee_r
GEON_TYPE : S_C_CO
POSITION
TRANS: (245.5 114.5 19.0)
ROTATION
RST: (4.9 -1.3 -0.3)
AX_SUP: 5.0
AY_SUP: 6.0
AZ_SUP: 7.5
EPSILON_1: 0.1
EPSILON_2: 0.5
TAPER_X_DEFORM: 0.05
TAPER_Y_DEFORM: 0.05
RADIUS_BENDING: 0.05

SUP_FEATURE : SUPERQUADRIC
NAME : knee_l
GEON_TYPE : S_C_CO
POSITION
TRANS: (245.5 75.0 19.0)
ROTATION
RST: (4.9 -1.3 -0.3)
AX_SUP: 5.0
AY_SUP: 6.0
AZ_SUP: 7.5
EPSILON_1: 0.1
```

```
EPSILON_2: 0.5
TAPER_X_DEFORM: 0.05
TAPER_Y_DEFORM: 0.05
RADIUS_BENDING: 15.0

SUP_FEATURE : SUPERQUADRIC
NAME : leg_r
GEON_TYPE : S_S_CO
POSITION
TRANS: (285.0 114.0 14.0)
ROTATION
RST: (3.1 -1.5 1.5)
AX_SUP: 7.0
AY_SUP: 14.5
AZ_SUP: 34.0
EPSILON_1: 0.1
EPSILON_2: 0.1
TAPER_X_DEFORM: 0.1
TAPER_Y_DEFORM: 0.1
RADIUS_BENDING: 15.0

SUP_FEATURE : SUPERQUADRIC
NAME : leg_l
GEON_TYPE : S_S_CO
POSITION
TRANS: (285.0 74.0 14.0)
ROTATION
RST: (3.1 -1.5 1.5)
AX_SUP: 7.0
AY_SUP: 14.5
AZ_SUP: 34.0
EPSILON_1: 0.1
EPSILON_2: 0.1
TAPER_X_DEFORM: 0.1
TAPER_Y_DEFORM: 0.1
RADIUS_BENDING: 15.0


SUP_FEATURE : SUPERQUADRIC
NAME : foot_r
GEON_TYPE : S_S_TA
POSITION
TRANS: (324.5 116.0 15.0)
ROTATION
RST: (0.4 -0.3 -2.0)
AX_SUP: 5.0
AY_SUP: 8.5
AZ_SUP: 14.5
```

```
EPSILON_1: 0.1
EPSILON_2: 0.1
TAPER_X_DEFORM: 0.4
TAPER_Y_DEFORM: 0.4
RADIUS_BENDING: 15.0

SUP_FEATURE : SUPERQUADRIC
NAME : foot_l
GEON_TYPE : S_S_TA
POSITION
TRANS: (324.0 74.5 15.5)
ROTATION
RST: (0.4 -0.3 -2.0)
AX_SUP: 5.0
AY_SUP: 8.5
AZ_SUP: 14.5
EPSILON_1: 0.1
EPSILON_2: 0.1
TAPER_X_DEFORM: 0.4
TAPER_Y_DEFORM: 0.4
RADIUS_BENDING: 15.0


ADJACENCY
NAME : head
SUPERQUADRIC : (neck)
ADJACENCY
NAME : neck
SUPERQUADRIC : (head chest)
ADJACENCY
NAME : chest
SUPERQUADRIC : (neck shoulder_r shoulder_l belly)
ADJACENCY
NAME : shoulder_r
SUPERQUADRIC : (chest upper_arm_r)
ADJACENCY
NAME : shoulder_l
SUPERQUADRIC : (chest upper_arm_l)
ADJACENCY
NAME : upper_arm_r
SUPERQUADRIC : (shoulder_r elbow_r)
ADJACENCY
NAME : upper_arm_l
SUPERQUADRIC : (shoulder_l elbow_l)
ADJACENCY
NAME : elbow_r
SUPERQUADRIC : (shoulder_r lower_arm_r)
ADJACENCY
```

```
NAME : elbow_l
SUPERQUADRIC : (shoulder_l lower_arm_l)
ADJACENCY
NAME : lower_arm_r
SUPERQUADRIC : (elbow_r hand_r)
ADJACENCY
NAME : lower_arm_l
SUPERQUADRIC : (elbow_l hand_l)
ADJACENCY
NAME : hand_r
SUPERQUADRIC : (lower_arm_r)
ADJACENCY
NAME : hand_l
SUPERQUADRIC : (lower_arm_l)
ADJACENCY
NAME : belly
SUPERQUADRIC : (chest hip)
ADJACENCY
NAME : hip
SUPERQUADRIC : (belly upper_thigh_r upper_thigh_l)
ADJACENCY
NAME : upper_thigh_r
SUPERQUADRIC : (hip thigh_r)
ADJACENCY
NAME : upper_thigh_l
SUPERQUADRIC : (hip thigh_l)
ADJACENCY
NAME : thigh_r
SUPERQUADRIC : (upper_thigh_r knee_r)
ADJACENCY
NAME : thigh_l
SUPERQUADRIC : (upper_thigh_l knee_l)
ADJACENCY
NAME : knee_r
SUPERQUADRIC : (thigh_r leg_r)
ADJACENCY
NAME : knee_l
SUPERQUADRIC : (thigh_l leg_l)
ADJACENCY
NAME : leg_r
SUPERQUADRIC : (knee_r foot_r)
ADJACENCY
NAME : leg_l
SUPERQUADRIC : (knee_l foot_l)
ADJACENCY
NAME : foot_r
SUPERQUADRIC : (leg_r)
ADJACENCY
```

```
NAME : foot_l
SUPERQUADRIC : (leg_l)

ARTICULATION
NAME : foot_r
SUPERQUADRIC : (leg_r)
DIST : 21.0                        % distance limit allowed for articulation
ARTICULATION
NAME : foot_l
SUPERQUADRIC : (leg_l)
DIST : 21.0
ARTICULATION
NAME : leg_r
SUPERQUADRIC : (knee_r foot_r)
DIST : 20.0
ARTICULATION
NAME : leg_l
SUPERQUADRIC : (knee_l foot_l)
DIST : 20.0
ARTICULATION
NAME : hand_r
SUPERQUADRIC : (lower_arm_r)
DIST : 21.0
ARTICULATION
NAME : hand_l
SUPERQUADRIC : (lower_arm_l)
DIST : 21.0
ARTICULATION
NAME : lower_arm_r
SUPERQUADRIC : (elbow_r hand_r)
DIST : 17.5
ARTICULATION
NAME : lower_arm_l
SUPERQUADRIC : (elbow_l hand_l)
DIST : 17.5
```

Appendix E

Quaternions

A quaternion is represented by a vector with four components, one real and three different imaginary parts. Let $\tilde{\mathbf{q}}$ be a quaternion, as:

$$\tilde{\mathbf{q}} = (q_0, \mathbf{q}) = (q_0, q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}) \tag{E.1}$$

An important operation with quaternions is multiplication, which in quaternion algebra is associative but not commutative. If,

$$\tilde{\mathbf{r}} = \tilde{\mathbf{p}}\tilde{\mathbf{q}} \tag{E.2}$$

where,

$$\begin{aligned} \tilde{\mathbf{r}} &= (r_0, \mathbf{r}) \\ \tilde{\mathbf{p}} &= (p_0, \mathbf{p}) \\ \tilde{\mathbf{q}} &= (q_0, \mathbf{q}) \end{aligned} \tag{E.3}$$

the scalar and vector componentsaref ˜

$$r_0 = p_0 q_0 - \mathbf{p} \cdot \mathbf{q} \tag{E.4}$$

$$\mathbf{r} = p_0\mathbf{q} + q_0\mathbf{p} + \mathbf{p} \times \mathbf{q} \tag{E.5}$$

Other useful operations with quaternions are:

- Dot product:

$$\tilde{\mathbf{p}} \cdot \tilde{\mathbf{q}} = p_0 q_0 + \mathbf{p} \cdot \mathbf{q} \tag{E.6}$$

- Norm of a quaternion:

$$\|\tilde{\mathbf{q}}\| = \tilde{\mathbf{q}} \cdot \tilde{\mathbf{q}} = q_0^2 + \mathbf{q} \cdot \mathbf{q} \tag{E.7}$$

A unit quaternion is then defined as: $\|\tilde{\mathbf{q}}\| = 1$

- A conjugate of a quaternion:

$$\tilde{\mathbf{q}}* = q_0 - \mathbf{q} \tag{E.8}$$

Unit quaternions can be used to represent rotations. If we have a rotation θ about the unit vector $\mathbf{w}$, a quaternion $\tilde{\mathbf{q}}$ which represents this is:

$$\tilde{\mathbf{q}} = \sin\frac{\theta}{2} + \mathbf{w}\cos\frac{\theta}{2} \tag{E.9}$$

Applying a rotation specified by $\tilde{\mathbf{q}}$ onto a vector $\mathbf{v}$ yields:

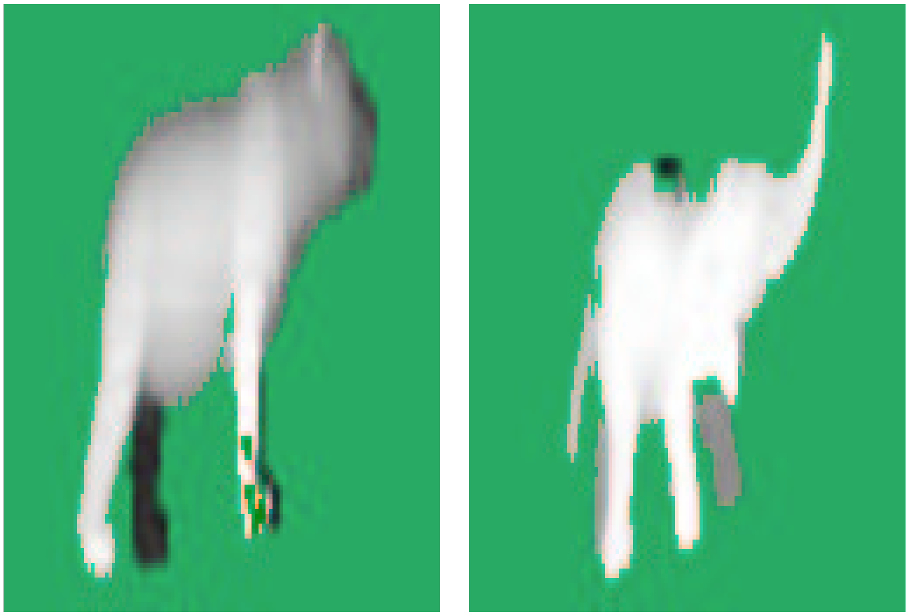
$$\tilde{\mathbf{v'}} = \tilde{\mathbf{q}}\tilde{\mathbf{v}}\tilde{\mathbf{q}}* \tag{E.10}$$

where $\tilde{\mathbf{v'}}$ is the quaternion representing the final rotated vector, *i.e.* $\tilde{\mathbf{v'}} = (0, \mathbf{v}')$, $\tilde{\mathbf{v}}$ is the quaternion of $\mathbf{v}$, *i.e.* $\tilde{\mathbf{v}} = (0, \mathbf{v})$, and $\tilde{\mathbf{q}}*$ is the conjugate of $\tilde{\mathbf{q}}$.

Appendix F

More Test Images and Results

This appendix includes a different set of thirteen (13) range images taken for further testing the recognition approach under more extreme circunstances. Figures F.1 to F.7 show the range images of the objects, eight (8) of these images are of two (2) objects but they were scanned from four different positions. Results after entering these range images into the recognition stages are shown as follows: Figure (F.8) to Figure (F.14) show the discontinuities found; Figure (F.15) to Figure (F.27) show the closed regions after segmentation; and Figure (F.28) to Figure (F.34) show the matched parts of the models on the final pose overlayed on the 3-D data.



(a) (b)

Figure F.1: Range images of objects (a) “bear”; (b) “elephant”

(a) (b)

Figure F.2: Range images of objects (a) “cheetah”; (b) “hippopotamus”

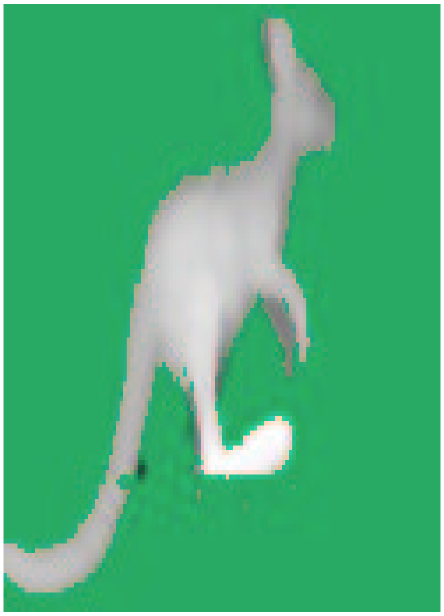
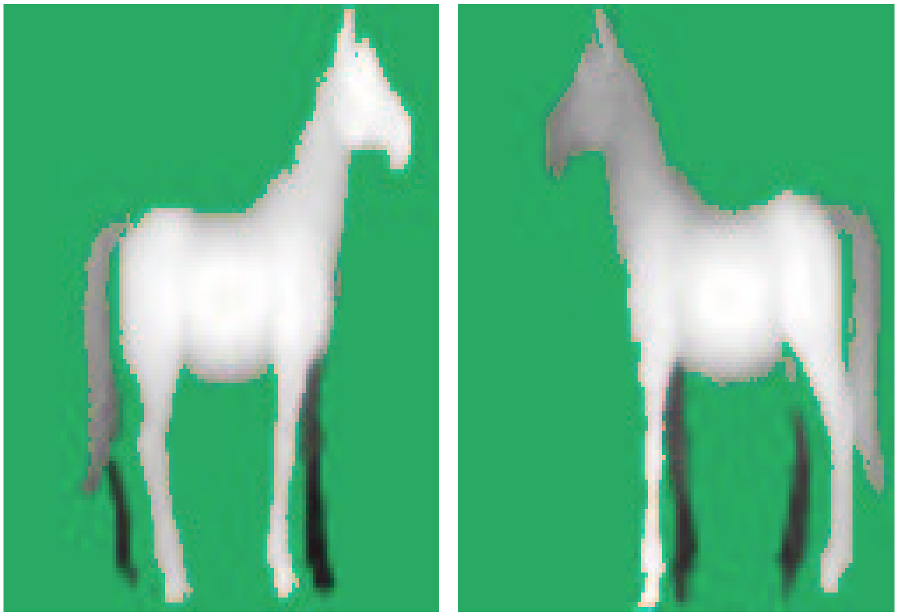
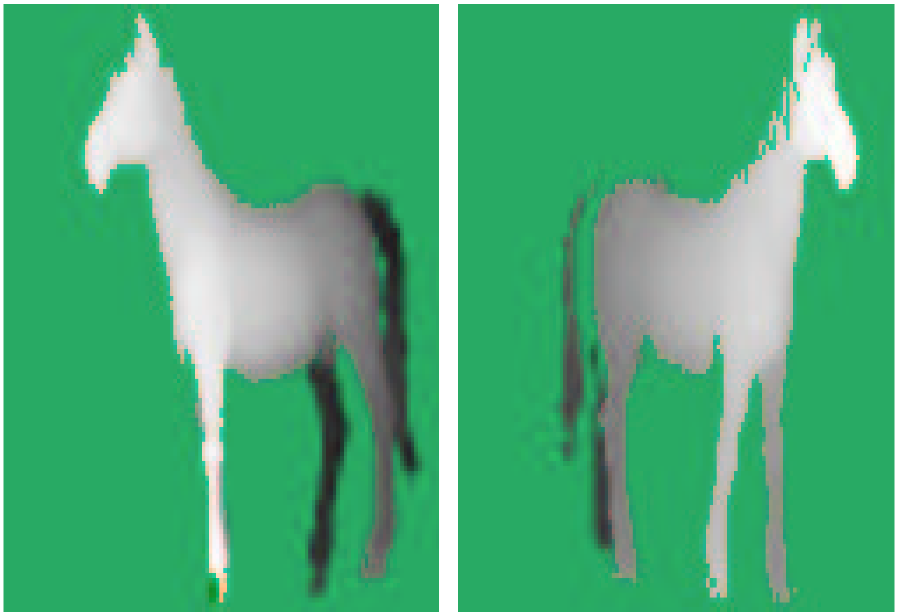


Figure F.3: Range image of object “kangaroo”



(a) (b)

Figure F.4: Range images of object "horse" in two (out of four) different positions



(a) (b)

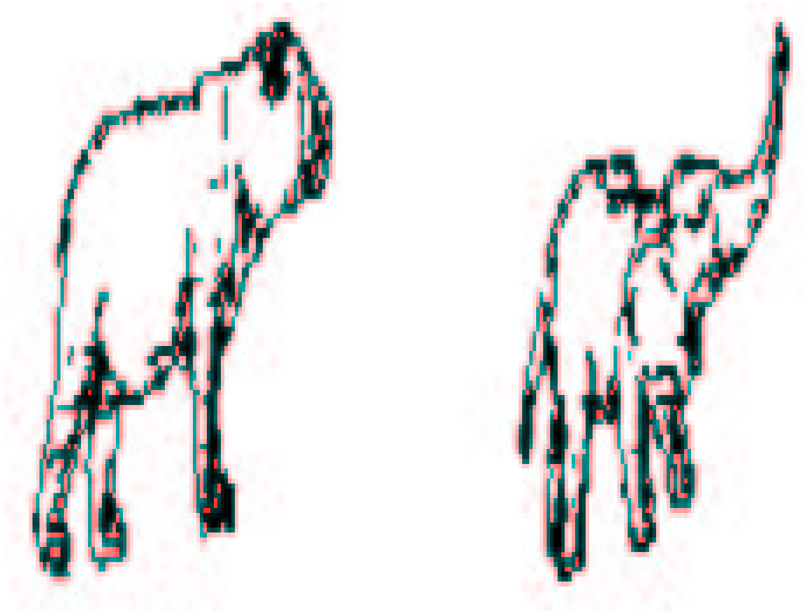
Figure F.5: Range images of object "horse" in two (out of four) different positions

(a) (b)

Figure F.6: Range images of object “pig” in two (out of four) different positions

(a) (b)

Figure F.7: Range images of object “pig” in two (out of four) different positions



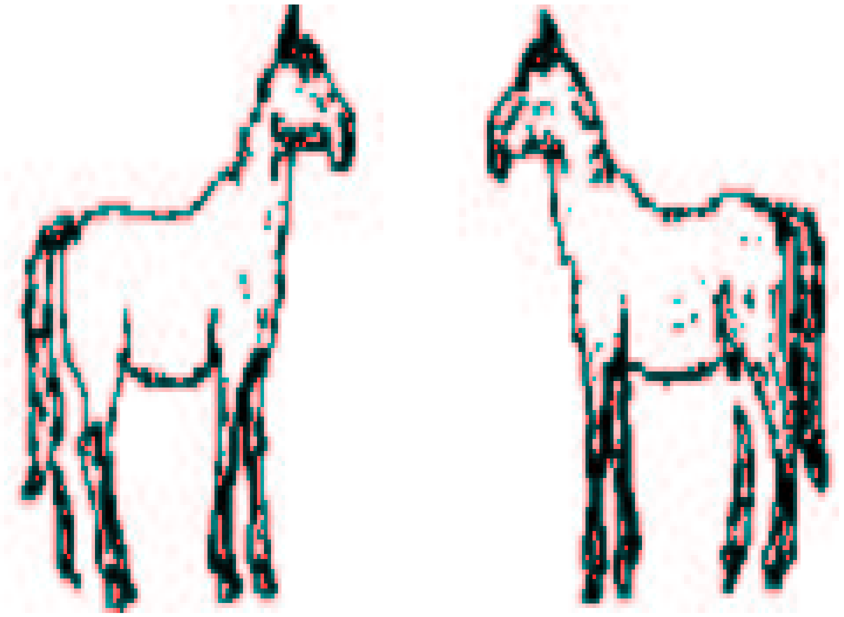
(a) (b)

Figure F.8: Depth, orientation, and curvature minima discontinuities found in the range images of objects (a) “bear” ; (b) “elephant”

(a) (b)

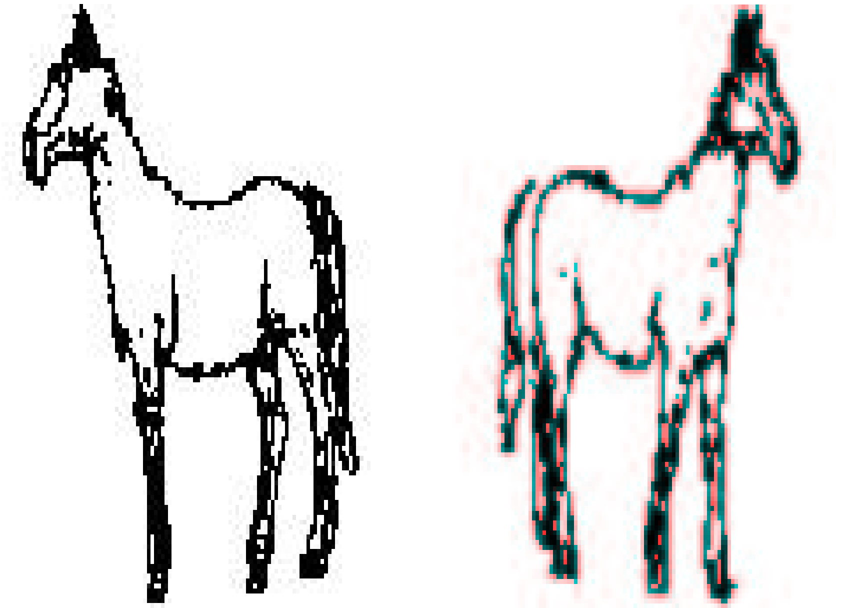
Figure F.9: Depth, orientation, and curvature minima discontinuities found in the range images of objects (a) “cheetah” ; (b) “hippopotamus”

Figure F.10: Depth, orientation, and curvature minima discontinuities found in the range image of object “kangaroo”



(a) (b)

Figure F.11: Depth, orientation, and curvature minima discontinuities found in the range images of object “horse” in two (out of four) different positions



(a) (b)

Figure F.12: Depth, orientation, and curvature minima discontinuities found in the range images of object “horse” in two (out of four) different positions

(a) (b)

Figure F.13: Depth, orientation, and curvature minima discontinuities found in the range images of object “pig” in two (out of four) different positions

(a) (b)

Figure F.14: Depth, orientation, and curvature minima discontinuities found in the range images of object “pig” in two (out of four) different positions

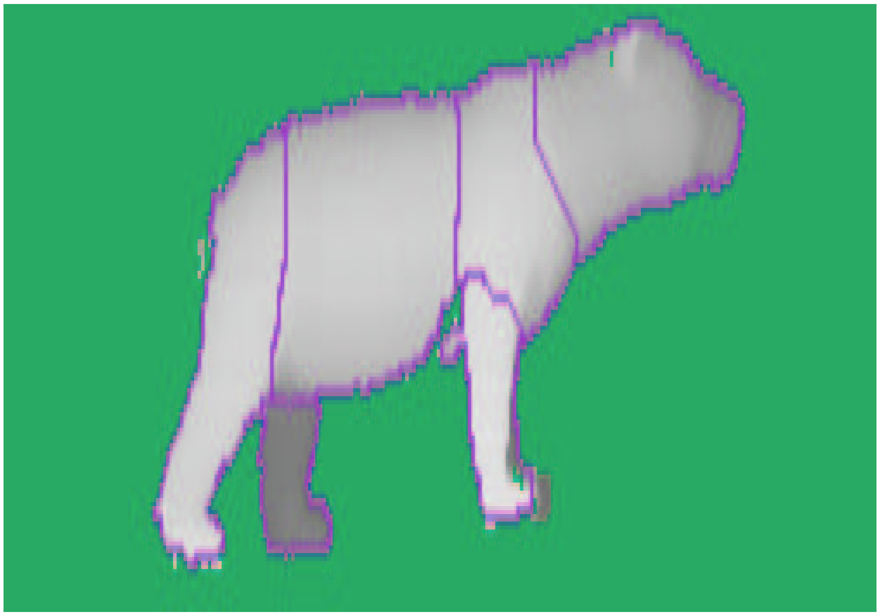


Figure F.15: Closed regions found by the segmentation stage for object “bear”

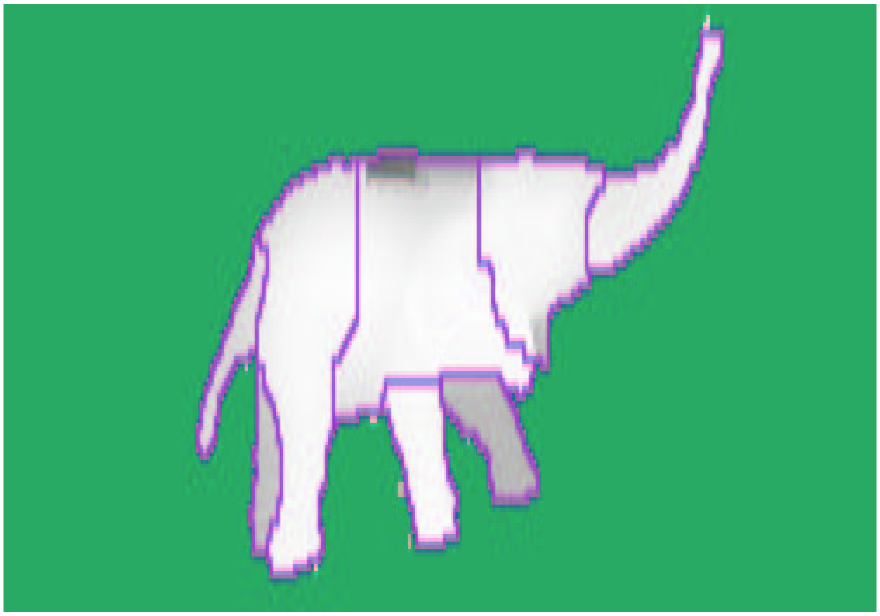


Figure F.16: Closed regions found by the segmentation stage for object “elephant”

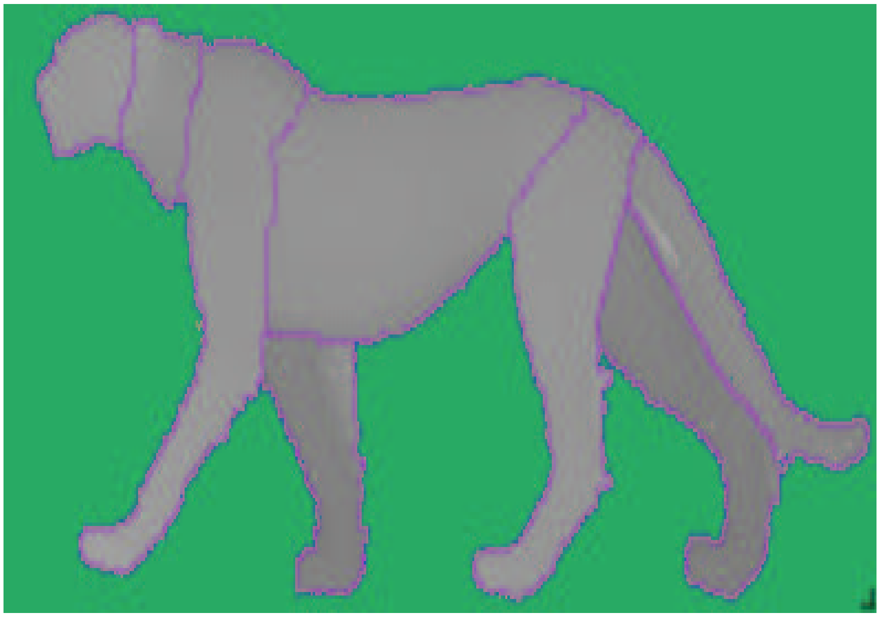


Figure F.17: Closed regions found by the segmentation stage for object “cheetah”

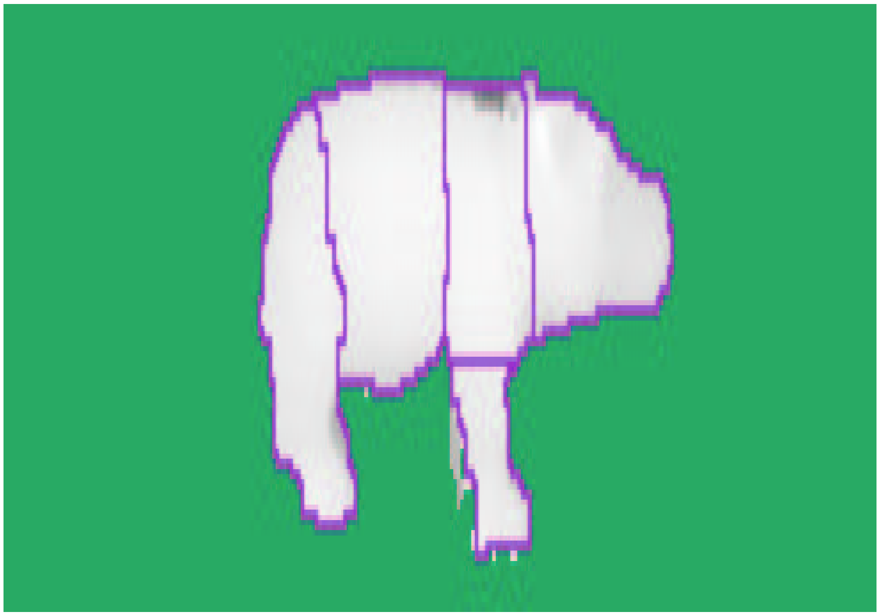


Figure F.18: Closed regions found by the segmentation stage for object “hippopotamus”

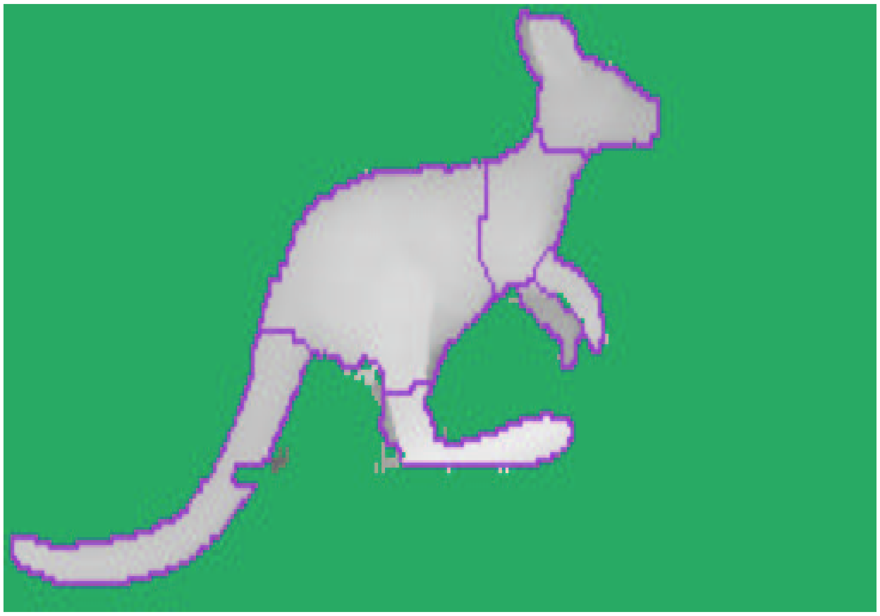


Figure F.19: Closed regions found by the segmentation stage for object “kangaroo”

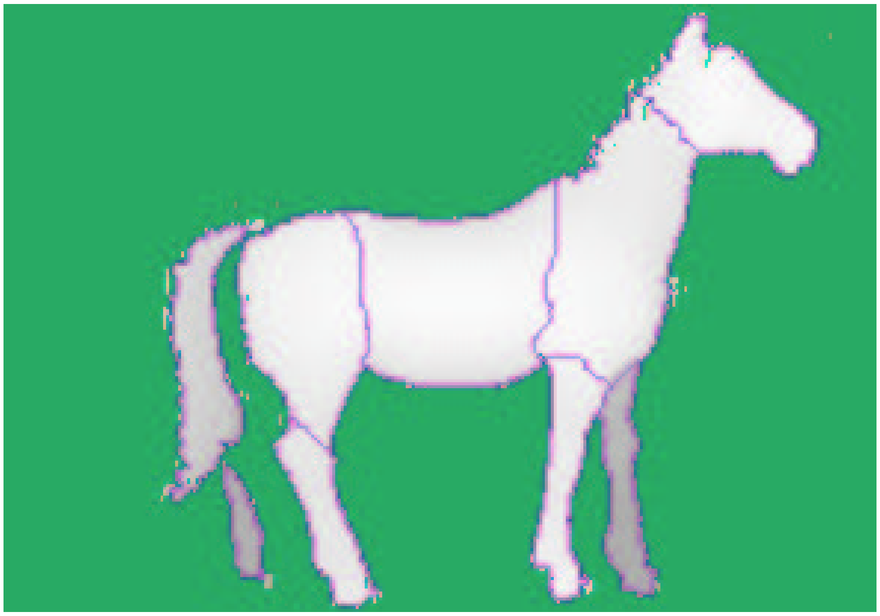


Figure F.20: Closed regions found by the segmentation stage for object “horse” in one (out of four) different positions

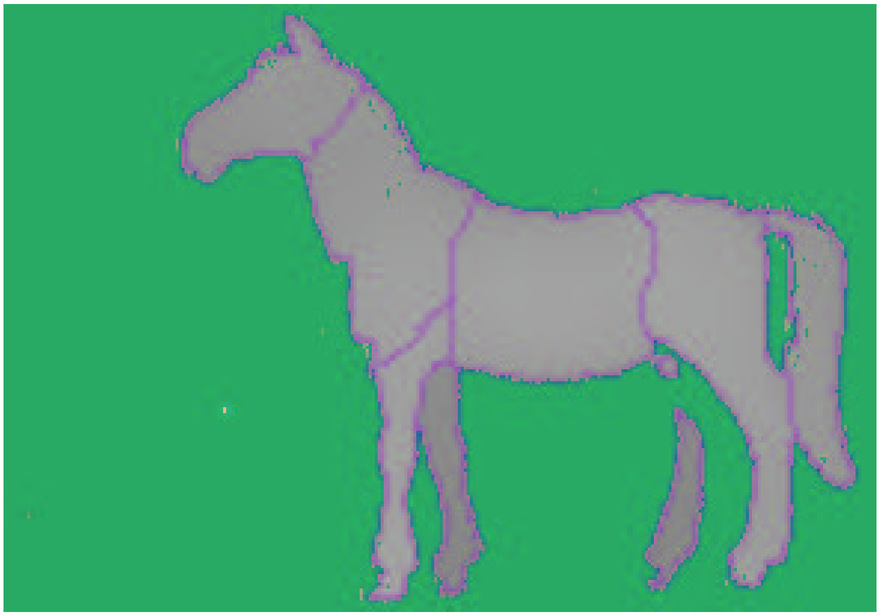


Figure F.21: Closed regions found by the segmentation stage for object "horse" in one (out of four) different positions

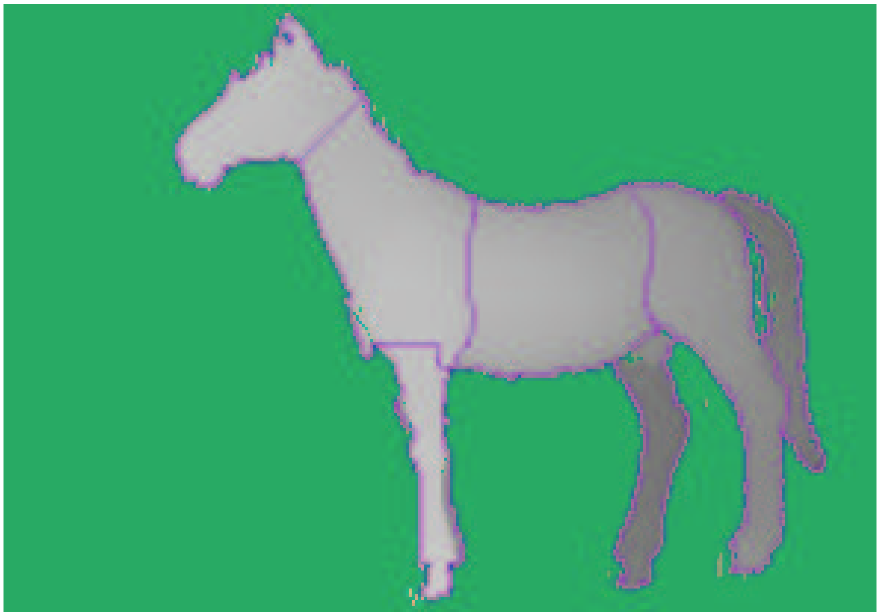


Figure F.22: Closed regions found by the segmentation stage for object "horse" in one (out of four) different positions

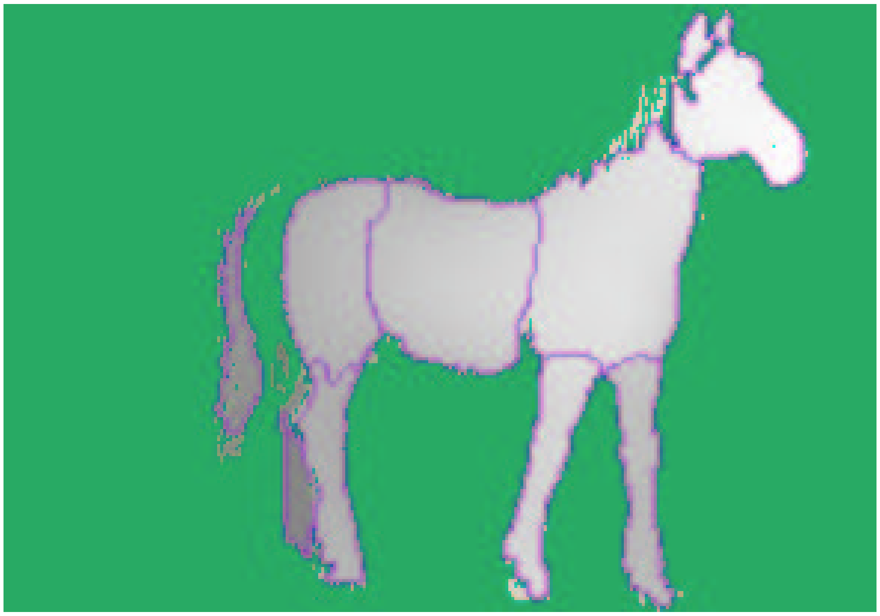


Figure F.23: Closed regions found by the segmentation stage for object “horse” in one (out of four) different positions

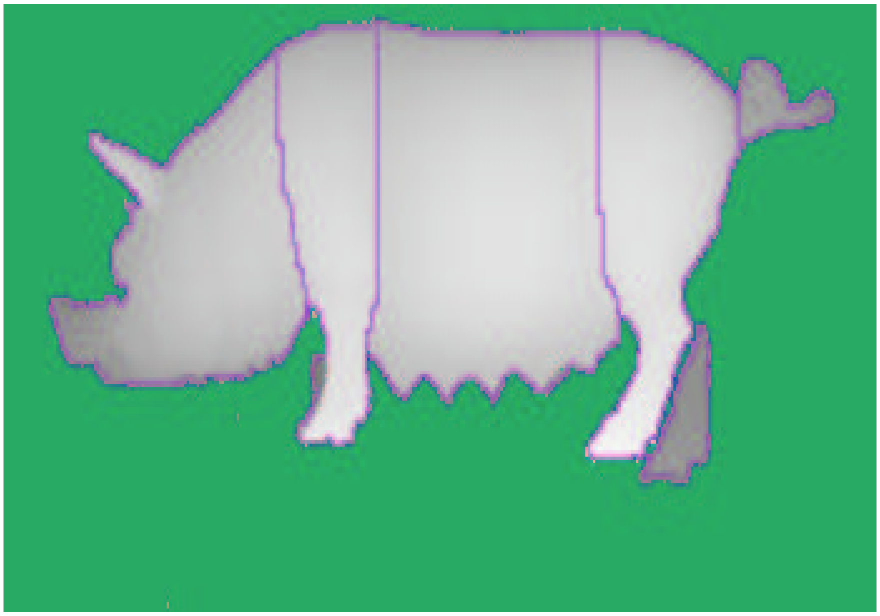


Figure F.24: Closed regions found by the segmentation stage for object “pig” in one (out of four) different positions

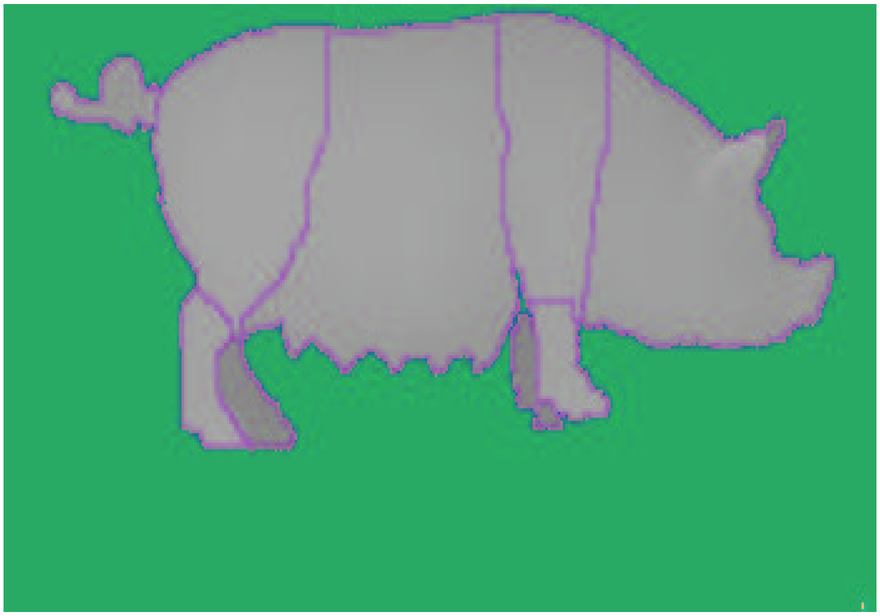


Figure F.25: Closed regions found by the segmentation stage for object "pig" in one (out of four) different positions

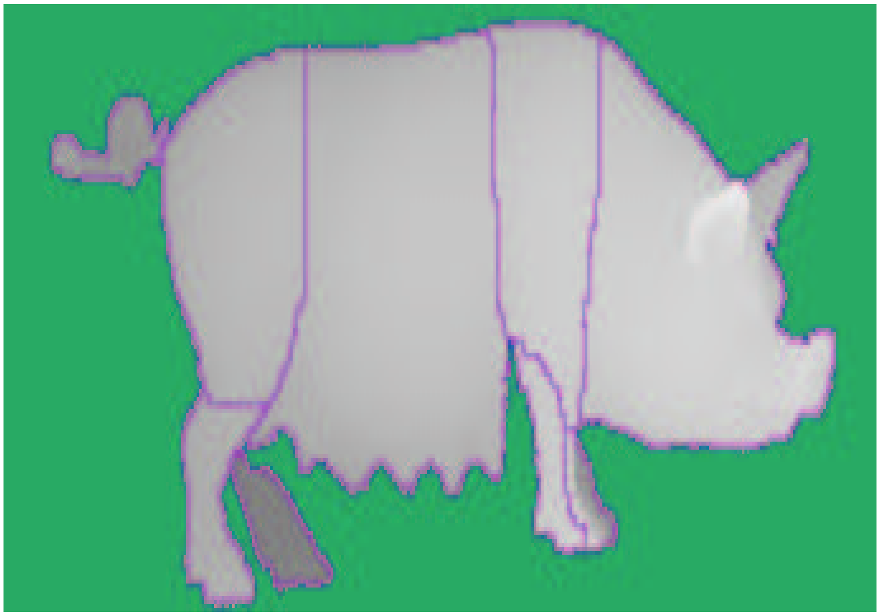


Figure F.26: Closed regions found by the segmentation stage for object "pig" in one (out of four) different positions

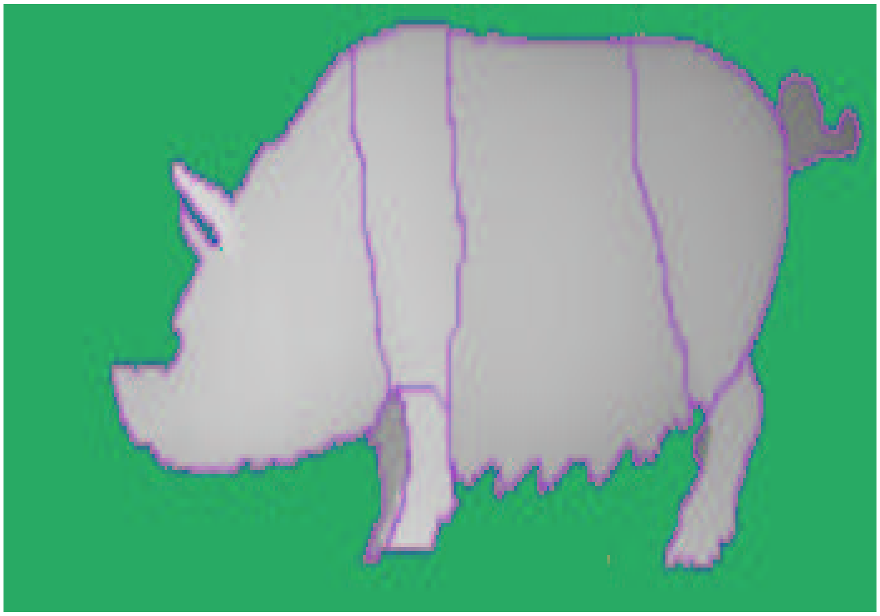
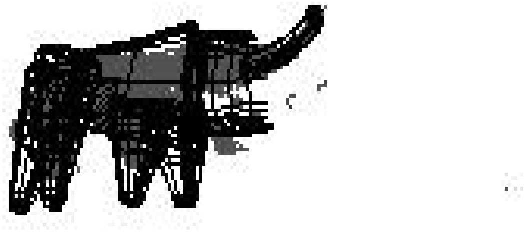


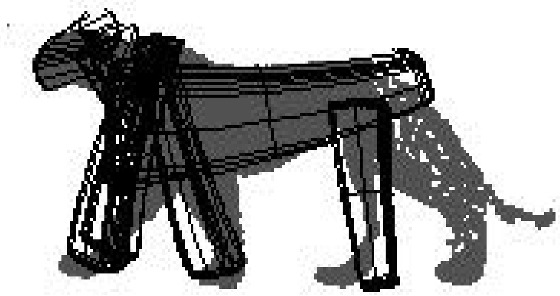
Figure F.27: Closed regions found by the segmentation stage for object "pig" in one (out of four) different positions

(a)

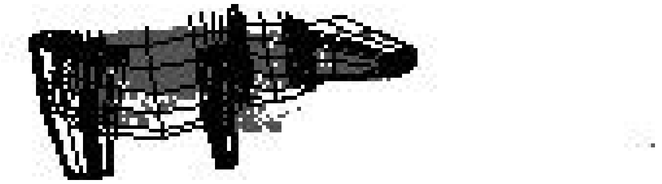


(b)

Figure F.28: Successfully recognized parts of objects (a) "bear" (b) "elephant", on final localization



(a)



(b)

Figure F.29: Successfully recognized parts of objects (a) “cheetah” (b) “hippopotamus”, on final localization

Figure F.30: Successfully recognized parts of object “kangaroo” on final localization

Figure F.31: Successfully recognized parts of object "horse" in two (out of four) different positions, on final localization

Figure F.32: Succesfully recognized parts of object “horse” in two (out of four) different positions, on final localization

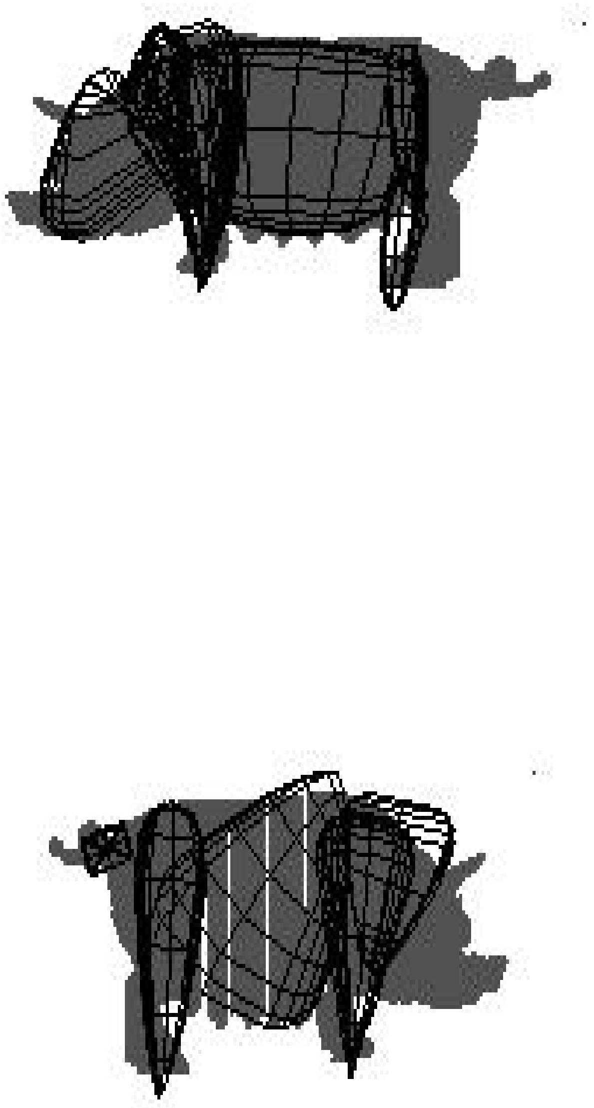


Figure F.33: Some recognized parts of object “pig” in two (out of four) different positions, on final localization

Figure F.34: Some recognized parts of object "pig" in two (out of four) different positions, on final localization