Recognition of Three Dimensional Objects Using Deformable Models

H.W.Hughes

Ph.D.

University of Edinburgh

1991

Abstract

This thesis considers the problem of efficiently identifying and locating instances of classes of three dimensional objects by matching them with a single generic model that represents that entire class of object. Since a member of an object class will normally differ from the prototype that represents the class, the approach used here is to allow the model to stretch, or deform, to fit the object.

The input image data may contain one or more objects of unknown identity and location, each of which is assumed to belong to a class of object for which there is a corresponding model. Both objects and models are represented by three dimensional surface data with the object data pre–segmented into surfaces of uniform curvature.

The process of deforming and matching the models to the object data is achieved in two stages. In the first stage combinations of object surfaces are formed and a search made for suitable object to model correspondences. Simple constraints are developed to reduce the search space to an acceptable size. When a correspondence is achieved, an initial estimate of the stretch required in the model is made and the model that contains those surfaces is selected for further matching. Because only those models for which there is evidence in the image are selected for further matching, the search space is further reduced. The second stage of the process involves taking those models selected in the first stage and performing a rigorous geometric search for any remaining model to object correspondences. As part of this process, the locations of the objects in the image are predicted and the deformation parameters refined as new correspondences are found. The location and deformation parameters provide further constraints for the geometric search, reducing the search space still more.

Recognition is demonstrated with a variety of objects, both synthetic and real, and the results discussed.

The use of deformable models in object recognition was found to be a good means by which to represent and match objects from classes showing three types of deformation – scale, stretch and small variation. The model deformation as formulated enabled the identity of the corresponding objects and their parameters of deformation to be determined with accuracy and efficiency.

Acknowledgements

I would like to thank the following for their help in the preparation of this thesis. My supervisors Dr. R.B.Fisher and Dr. J.C.T.Hallam for their suggestions, comments and criticisms and Prof. J.A.M.Howe for allowing the use of the department's facilities when I was no longer funded. Particular thanks is due to Dr. Mike Cameron–Jones whose help and wit was appreciated on many occasions.

I would like to thank my parents for the financial support and the long term loan of their garden shed. My brother Trevor for the encouragement, the occasional diving trip and the more than occasional meal. Thanks especially to my friend Alison who, even after four years, is still a friend and for giving up her time, when I am sure she would much rather be doing something else, to do the interminable proof reading.

My acknowledgements would not be complete without mentioning the university canoe and sub–aqua clubs. Whilst it might be thought that the contribution of these to the field of three dimensional object recognition is somewhat limited, they do provide the opportunity to relax, clear the mind and approach the forthcoming Monday refreshed in spirit – if not in body! Indeed there is nothing like the river Orchy in spate for putting that niggling segmentation fault (again) in perspective or surviving the Oetz (at any level) to sort out the intricacies of dynamic array allocation. In recent months diving with the sub–aqua club has given me the opportunity to mull over a few ideas on the thesis in more tranquil surroundings. Hanging off the decompression line six metres down is perhaps not the most conventional place to compose a chapter but it is certainly quiet and something is necessary to distract the mind from minor things like low air levels and leaky neck seals.

I must mention my flatmates both past and present, especially Matthew, better known as Dr. Wave, and "The Greenmantle" which I have visited so often that I must have interests in Scottish and Newcastle.

Finally, I would like to thank my fellow students and colleagues from the Forrest Hill user's room and vision lab. for their comments (well, some of their comments) and advice. Myra Wilson and Piak Chongstitvatana deserve special mention, despite, or possibly because of, working for the robotics group, for putting up with and perhaps even listening to my ramblings, curses and awful jokes over the four years that we have "shared" an office.

This work was funded by the Science and Engineering Research Council.

Declaration

I declare that this thesis has been composed by myself and that the work described in it, is my own.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 An Introduction to Object Recognition

In this thesis, the *basic* form of object recognition is defined as the process whereby the features of one or more objects of unknown identity and position in an image are both identified and their positions, relative to some reference point, used to infer the identities and positions of the objects to which they belong.

Object identification is normally achieved by comparing or matching the visible features of the objects with the features of a set of stored models. Where a match is achieved that is, by some appropriate criteria, sufficiently good then there is said to be a correspondence between the object feature and that of the model. If a sufficient number of correspondences are obtained between a model and an object, then the identity of the object is assumed to be that represented by the model.

The position of an object is normally determined concurrently with the identification process. The stored models are usually defined in terms of the positions of their features within their own coordinate system, referred to here as the model coordinate frame. While the positions of the objects themselves are unknown, the positions of their individual features can be obtained from the image data according to a separate coordinate system, the world coordinate frame. The world coordinate frame is often, but not always, based on the imaging device. To estimate the position of an object, it is necessary to estimate the geometric transformation, known as the coordinate frame transformation, that will map the model in the model coordinate frame to the corresponding object in the world coordinate frame.

This basic form of object recognition is used extensively but is only applicable where the objects have a *rigid* structure, enabling the coordinate frame transformation to be calculated, and each is represented by its own model.

This approach is rather limiting for two reasons. Firstly, a large number of objects have features in a non–rigid structure and it is not possible to match these using standard rigid coordinate frame transformations. Secondly, different examples of a class of objects, although often very similar, will require different models which increases the size of the matching problem.

A solution to this problem, and the one described in this thesis, is to use *parameterised models*. The parameters chosen represent the features of the object that can change, other than its position in the image. By careful selection of the parameters to reflect the differences exhibited, one parameterised model can be used to represent an entire class of objects.

The basic definition of object recognition given earlier must now be extended. Parameterised object recognition is the process whereby the features of one or more objects of unknown identity and position in an image are identified and their positions *and parameters* determined.

1.2 Achieving Object Recognition

Having defined the process of object recognition how is it to be achieved? The first problem that must be approached is that of constructing the set of stored models. While it is possible with modern graphics techniques to produce visually realistic representations of almost any object, this is quite different to modelling an object for recognition. When modelling for recognition, the emphasis is less on producing a likeness and more on representing the relevant features in a way that they are easily extractable for direct matching with the types of features that can be extracted from the image. A related problem is that of representing the objects in the image. As with the modelled representations the visual appearance in the image of the object is of little immediate use. The samples obtained by the imaging device will be large in number, contain relatively little information about the object's shape and be individually susceptible to noise and illumination gradients. The image data must be processed to yield pertinent object features that are stable as regards error. Since these reflect the structure of the object, rather than the sampling arrangement of the imaging device, they provide a better basis for the matching problem.

What constitutes a pertinent object feature, which model features are to be modelled and the basic building blocks, or primitives, that are used to represent them is chosen by the designer, but is normally influenced by the types of object to be recognised and the image data available. For example, while the colouring of an object might be distinctive, it is of little use if the object is back lit to form a silhouette. Similarly, defining the model in terms of volumetric primitives would be inappropriate.

Having arrived at some suitable representation for both the objects in the image and the stored models, the next problem to be addressed is that of object identification. Identification is achieved by finding correspondences between the features of the objects in the image and features in the stored models.

The naive approach is to generate all possible combinations of *object to model* matches. If each object in the image corresponds to one of the stored models then one or more of these combinations will be correct, although the number of combinations generated will usually be very large. Each combination of object to model correspondences proposes a hypothesis of the object's identity and, eventually, its position. If the correct hypotheses can be found then the objects in the image will have been identified.

This naive approach is not without its pitfalls. The number of possible combinations can often be so large as to make the task impractical in terms of the effort required, though solutions have been devised to overcome this. A further problem concerns the accuracy of the identification process. Often, only some of an object's features are immediately identifiable in the image and if more than one model represents these features then an unambiguous identification is not possible.

An alternative, but equally naive, approach is to generate all possible combinations of *model to object* correspondences. Since correspondences are sought for all model features, it is usually possible to obtain an unambiguous identification. However, since the position of the corresponding object is unknown (at least initially), it is not possible to determine which object features are expected to be visible in the image and so which model features should have a correspondence. The number of possible combinations is again very large but, as before, there are methods to reduce this figure.

Virtually all existing recognition systems use one or both of these approaches as the basis of their operation. Which method is used depends largely on the type of image data available and how much is know in advance about the objects in the image.

In its simplest form, object identification involves matching *rigid* two dimensional objects, such as templates, with two dimensional models that are exact representations of the original object. Since the objects are laminar in form, the hiding, or occlusion, of one feature by another feature of the same object (self occlusion) will not occur and, unless occluded by another object, all object features should be visible in the image. Generating object to model and model to object correspondences is relatively easy and provided the correct hypotheses can be found, identification is straightforward.

Estimating the position of a two dimensional object in an image is relatively easy. Since such objects generally lie on a flat surface, the position of the object is restricted to a translation in the two orthogonal directions across the surface giving two translational degrees of freedom, and a rotation about an axis normal to the surface giving a rotational degree of freedom. Since all the object features are normally visible, as the correspondences between model and object or object and model are established then an accurate estimate of the rigid geometric transformation between model and object can be established by simple two dimensional geometry.

A more complex problem of recognition involves a two dimensional image of three dimensional objects to be matched with a set of three dimensional models. Because the far side of any three dimensional object relative to the imaging device will not be visible in the image, some of the object features will always be self occluded. Which features are hidden depends on the orientation of the object in the image, and as the orientation of an object changes, so too does its visual appearance. The naive approach discussed earlier of generating all combinations of object to model matches works well provided that there are enough features visible for each object to be identified unambiguously. Since some of the object features are missing, trying to match all the features represented in a model is doomed to failure. For this approach to work, it is necessary that the orientation of the object be known, at least approximately, and this usually requires that the identity of the object be known too. If the orientation of the object is known, the model can be subjected to the same rotation and the visibility of its features predicted. Since the matching problem now becomes one of finding matches for those model features that are not self occluded, it is relatively efficient.

If the object is rigid then its position in the image now has six degrees of freedom, three translational along the three orthogonal axes of the world coordinate frame and three rotational degrees of freedom about these axes. While the orientation of the object and any translation orthogonal to the axis of the imaging device can be recovered from the image, the translation along this axis, known as the range or depth, is lost during the projection process and must be inferred from visual cues such as shading, texture or stereo.

A way of overcoming the restrictions of two dimensional projections of three dimensional objects is to use a three dimensional image of the scene. A three dimensional image represents not the visual appearance of the object but instead, at each point in the image, the range to the nearest solid object is sampled. The process of object identification remains as before, but the estimation of the object's position is simplified since all three components of any translation are directly available from the image data rather than having to be inferred.

1.3 Some Problems Associated with Object Recognition

One of the main problems common to both the approaches described previously is that of combinatorial explosion. As the number of model features represented or the number of object features in the image increases, the number of possible combinations of matches increases exponentially. The number of combinations quickly becomes so great that even for a database of relatively few models and an image containing a single object, it becomes impractical to generate them all.

Assuming that all the combinations can be generated, the next problem is that of determining which of them are correct. Ideally, there will be only one correct hypothesis for each object in the image, but these must be correctly identified from the many millions of possible combinations. If the models are exact representations of the corresponding objects then a rigorous geometric comparison between each model feature and each object feature in a combination should eliminate all the incorrect combinations but would require a huge computational effort. The use of more elementary checks where a match between model and image fulfilled certain simple conditions or constraints would reduce the effort required but at the risk of reduced accuracy.

A further problem is that of object representation. If the models are exact replicas of the original objects then each time a new object is to be recognised, a new model is required. This will not only require the effort involved in the construction of a new model, but will increase the combinatorial problem mentioned above. This approach also lacks flexibility in that similar objects have completely separate models. There is no representation of a group or class of objects with which a new object with no corresponding model could be classified as a member of that set.

Many solutions have been proposed to overcome these problems. The success to which they achieve this depends largely on the image data available and what information is known about the object represented by the model.

To reduce the number of combinations to a manageable level, several techniques can be used.

One is that of using simple geometric constraints to identify incorrect correspondences at an early stage in the analysis or search of the combinations generated. If an incorrect correspondence can be identified then all the combinations of which it forms a part may be eliminated. The earlier in the search that this can be achieved, the smaller the number of combinations that need be checked.

A second approach is to split both model and object up into conceptual parts. The combinatorial problem is reduced to that of matching the features of an object part to those of a model part and then using combinations of model parts to identify the object as a whole. Unfortunately this approach then has the added complexity of attempting to identify which model parts should be matched with the image.

A further approach is to initially disregard much of the image data and quickly search the image for distinctive object features. Where an object feature can be uniquely identified, only the corresponding model or model part need be matched with the image.

A final approach is to reduce the number of models by the use of a single model to represent a class, or family, of objects. This is usually done by including in the model variables, or parameters, which represent the differences found between individual examples of that class. Unfortunately the recognition process must now not only establish the identity and position of the object, but must also estimate the values of the model parameters for the particular object in question.

1.4 Parameterised Object Recognition

Using parameterised models to represent classes of objects rather than individual examples has several advantages. Apart from the smaller number of models required, if a previously unseen object is to be recognised then provided it is a member of an existing class there is no requirement to construct a new model.

The problem can be broken down into three distinct areas:

- Firstly, there is the problem of object representation. How can a whole class of objects be represented by a single generic model? Such a model must be capable of allowing for the differences between individual examples of a class while representing the features of that class of object with sufficient accuracy for reliable matching.
- Secondly, faced with a number of features in the image, how are these to be reliably matched with the features of the stored models when the features as represented by the generic model may differ greatly from the corresponding features in the individual object?
- Finally, because the structure of the object may be different than that of the corresponding model a simple transformation, in terms of rotation and translation, from the object coordinate frame to the model coordinate frame is no longer possible. To determine the position of an object and to predict the position and appearance of its features, it is first necessary to establish in what way the model differs from the object (i.e. what values the model parameters must take) before the object to model transformation can be determined.

1.5 The Problem Approached in this Thesis

In this thesis the images used are three dimensional, or range data, images containing one or more objects of unknown identity and position. Each object in the image is expected to belong to one of the object classes modelled. As an instance of a class, each object may show individual differences of limited types but of unknown degree, relative to the generic class model.

The range data has already been segmented into surface regions or patches and because the parameters of the imaging device are known then these surfaces can be defined in terms of absolute curvatures and distances. It is this object surface data that forms the input to the problem.

The problem is to *efficiently identify each object* as belonging to a particular class (represented by a generic class model) from their surface data, to *determine the position* of each object from the positions of their surface data and to *determine the parameters* of each object relative to the corresponding class model. The problem is *not* to uniquely recognise each object, although its identity is implicit in the values calculated for the parameters.

1.6 Solving the Problem by Stretching a Point

For convenience, the images used in this thesis are three dimensional range images, though the techniques described can be applied to any image data where surface data of an object is known or can be inferred. Given accurate three dimensional object surface data some form of segmentation is required to split the surface data into regions, or patches.

In this thesis the problem of parameterised object recognition is approached by the use of models that stretch or deform. To aid the matching process the basic primitives used to construct the models are also surface patches.

The generic models described in this thesis exhibit three types of deformation:

- Global scale – the *whole object* shows a uniform change in size along all its dimensions.

- Part stretch – a *part* of the object shows a change in size along up to three orthogonal directions; the part appearing "stretched". A part that is stretched may itself consist of parts that may also be stretched. The degree to which a part is stretched is individual to that part; other parts of the same object may show different degrees and directions of stretch.

- Individual variation – an *individual feature* of the object shows small changes in appearance and position not accounted for by the effects of the scale of the object or the stretch of the part of which it is a component.

Having built the set of generic models and having defined the rôle of their parameters, the problem now is to match them with the object features in the image.

The first stage of the recognition process is that of *model selection*. The aim of model selection is to *quickly* identify those models for which there is evidence that the corresponding object is present in the image. Since an object surface in isolation yields little information as to its identity, configurations of three object patches are generated. A simple tree search is used to quickly examine each possible combination of *object to model* correspondences. If the three object surfaces are assumed to match the same model part and in turn the same model, then they will be subject to the same degree of scale and stretch. This enables some simple constraints to be used to reduce the number of possible combinations of matches. Only those model parts which meet the constraints imposed are selected from the database for further matching. Techniques are developed which enable an initial estimate of the scale and stretch parameters to be determined without needing to know the model to world coordinate frame transformation.

The second stage of the process is that of *model verification*. Having selected a relatively small number of model parts for further analysis, these are deformed by the scale and stretch parameters determined previously. Now that the configuration of the model features should match those of the object, a first estimate of the model to world coordinate frame transformation can be determined. Once this is known the identity of the object part in the image can be verified by searching for any remaining *model to object* correspondences. As new correspondences are found the initial estimates of scale, stretch and individual variation can be refined and the coordinate frame transformation determined to greater accuracy.

Once an object part has been identified, the model selection process selects the larger model part of which it forms part and suggests that for verification. Where several parts belonging to the same class of object are identified, a simple tree search is used to determine if they could be parts of the same object. If this is the case, a first attempt is made to separate the stretch of the individual parts from the scale of the object as a whole.

The model verification process proceeds as before, but now the intention is to seek correspondences with model parts rather than model features. As new correspondences are found, the deformation and position parameters are refined further.

The model selection and verification processes proceed alternately in a generate and test fashion until eventually the models themselves are selected and verified and recognition is completed.

The techniques described in this thesis are applied to a number of images with a range of objects showing varying types and degrees of deformation. The accuracy of the results and the efficiency of the two processes are discussed and compared with previous work.

1.7 The Structure of the Thesis

Chapter two, the literature review, is a critical review of the existing literature on both three dimensional and, where applicable, two dimensional object recognition. Research that deals with the representation and recognition of parameterised, or deformable, objects is examined in particular detail.

Chapter three describes the three dimensional models used to represent classes of three dimensional objects and the deformation types allowed within a class. The global scale of an object, the stretch of an object part and the individual variation of an object feature are defined and their effects on the appearance of an object and the relative positions of its parts are discussed. Finally, the deformations are defined in terms of the geometric operations necessary to reproduce the same deformations in the model before matching can take place.

Chapter four concerns the process by which the deformation parameters of an object in the image are resolved. Since the three types of deformation identified in this thesis usually occur in combination, this chapter discusses how the effects of the different types of deformation can be separated and estimates of each obtained as part of the recognition process.

Chapter five describes how the pertinent features of each object in the image are recovered from the image data to form an object description suitable for matching directly to the model. The method by which the raw range data is first obtained and then segmented into surface patches corresponding to object features is described. Finally the process whereby the parameters of each surface patch are established is explained.

Chapter six describes the two sections of the model selection process. The first stage deals with the initial selection from the model database of model parts based on the evidence obtained from configurations of surface patches in the image and establishing first estimates of the deformation parameters for those object parts. The second stage of the selection process concerns the recursive selection of superparts or whole models based on the verified existence in the image of their constituent parts and the successive resolution and refinement of the deformation parameters. In both sections the results and efficiency of the selection process on a number of images are discussed.

Chapter seven, the model verification chapter, is also in two sections. The first section explains the method by which the proposed existence of an object part in the image is either confirmed or denied by the use of a simple tree search to account for any unmatched features in the corresponding part model. As the verification process proceeds, the estimates of both the coordinate frame transformation and the deformation parameters are continually refined as new model to object correspondences are discovered. The second section is similar and describes the method by which the existence of a superpart or model is confirmed by searching the image for any unmatched model parts and, if appropriate, updating the position estimate further. The results and efficiency of the verification process when applied to several images are discussed.

Chapter eight describes in detail the numerical algorithms and techniques used during object identification for the estimation of an object's position, scale, stretch and variation parameters.

Chapter nine summarises what has been achieved, the shortcomings of the system and proposed extensions.

Chapter 2

Literature Review

In this chapter, previous research in the fields of object representation and three dimensional object recognition is summarised and the shortcomings which motivated the work described in this thesis are discussed.

2.1 Object Representation

2.1.1 Marr's Criteria for Object Representation

Marr [Marr, 1982] reviewed the problem of object representation for computer recognition and proposed the five following criteria.

1. Accessibility – The information required by the recognition process should be readily available in the representation rather than having to be derived with the use of large amounts of memory or processing.

2. Scope – It should be possible to model the object with the facilities available in the representation. I.e. to model a sphere would require a representation in which spherical primitives were available.

3. Uniqueness – It should be possible to represent an object uniquely.

4. Stability – A similarity in the structure of two objects should be reflected by a similarity in the structure of their respective models.

5. Sensitivity – The representation should be sensitive enough to reflect the small differences that differentiate two similar objects.

The ideal modelling system therefore should aim to meet these criteria. Unfortunately this problem is by no means easy and becomes progressively harder as the objects to be modelled become more complex.

2.1.2 Modelling Using Volumes

Brooks [Brooks, 1983] used generalised cones as the modelling primitives for object representation in his ACRONYM recognition system. Objects were constructed by specifying the translation and orientation of each generalised cone in terms of the coordinate frame of the object. These objects and larger primitives could in turn be used to form larger objects. The models produced in this way had a hierarchical tree structure. This enabled the object to be represented in a coarse to fine manner with different levels of the model tree representing the object to different levels of detail.

The models used by the ACRONYM system attempted to model not only individual examples of objects but also large generic classes of object by allowing for the variations found between different objects of the same class. Three distinct types of variation were represented. These were variations in the size, structure and spatial (position and orientation) relationships of the represented objects. In the simplest form variations were represented by the use of free variables rather than constants in the model specification. Alternatively, rather than allowing a model variable to take any value, limits, or constraints, could be used to specify an acceptable range of variation. A generic class of objects could be represented by a model with a carefully chosen set of constraints. The addition of further or tighter constraints enabled subclasses of objects to be defined, while still further constraints allowed models of specific instances of the class to be represented. In this way a single model could represent both a generic class of object as well as an individual example.

This system was extremely powerful although the use of volumetric primitives meant that for the recognition of two dimensional images some processing was required before model and object primitives could be matched. A further problem was the rather awkward representation of changes in structure.

Pentland [Pentland, 1985] [Pentland, 1986] showed that existing modelling systems, that had often originally been developed for fields such as physics or engineering, had many shortcomings with regards to their use in constructing models for computer recognition. Models that were exact replicas of the original object were very complex while impoverished representations offered reduced reliability and discrimination in the recognition process.

Pentland proposed that the world could be modelled by intermediate grain models using a small set of processes that occurred again and again with the more complex shapes being produced by compound combinations of these processes. His approach was to describe an object at a scale corresponding to the notion of an object "part". A part model had to be complex enough to offer reliable recognition but simple enough to serve as the building blocks, or primitives, for models of specific objects.

The suggested solution was to model the object part as if it had been notionally moulded from a "lump of clay". For this purpose superquadrics [Barr, 1981] were used as a modelling primitive that could be shaped or deformed. These basic lumps of clay formed the basis, or prototypes, of the modelling system. These prototypes were then acted upon by the processes of stretching, bending, twisting and tapering. The results could then be combined using Boolean operations into new, more complex prototypes. These, in turn, could be subjected to further deformation and combination enabling complex shapes to be constructed from a small number of prototypes and processes.

Bajcsy and Solina [Bajcsy & Solina, 1987] reviewed the way in which the human visual system appears to categorise objects and came to the following conclusions:

1. The human visual system is capable of differentiating between those object features that are incidental and those that are essential. The way in which such feature instances are sorted out is a form of categorisation. Hence objects are not

categorised arbitrarily but in a way that reflects the true structure of the world. A category is represented by a single prototype model that represents all the features or properties characteristic of that category. These characteristic features are held nowhere else, individual members within a category are specified in terms of, or by deviations from, the prototype model.

2. Categories are split into three levels. At the highest level are the supercategories. These tend to reflect an object's functional features (i.e. a supercategory might contain all vessels used for drinking). Because only functional features are represented, these categories are easier to recognise when the objects are in use than in static scenes. Basic categories are those that reflect the structure of the world (i.e. a cup and a mug would fall into different though similar basic categories). They have a high level of abstraction. Basic categories are the preferred level of reference, they are learned by children first and are perceived faster than sub or supercategories. Basic categories are divided into subcategories by only a few, often only one, perceptual or functional feature (i.e. differentiation between cups by colour, size of handle, etc.). While sub or supercategories often overlap in their descriptions, basic categories are mutually exclusive.

3. An object should be modelled as the prototype of a basic category. Sub and supercategory features are derived from this. The modelling of the basic prototype or generic model is not a simple problem. The representation must allow for the variations of shape found within a category and yet be able to differentiate between different basic categories. During recognition the prototype model will be compared with the object under consideration. To achieve equality between model and object, shape deformation is required.

In their choice of modelling primitives, Bajcsy and Solina borrowed heavily from Pentland's work. The basic building blocks were superquadrics that could be deformed and then merged using Boolean operations. A hierarchical model was proposed where the position of a part was specified in the level above. Allowing the position to vary allowed for variation in structure, while allowing the deformation parameters to vary within set limits allowed for the shape variations within the category.

Models built in this way were successfully used in the recognition of particular types of relatively simple classes of objects. They were of less use with more complex objects that were not well described by the lumps of clay prototype.

2.1.3 Modelling Using Surfaces

In the fields of computer aided design and computer graphics it is often necessary to *realistically* represent surfaces. The technique often used is that of the B–spline and the Coons surface patch [Gordon & Riesenfeld, 1974] [Coons, 1974]. The B–spline is a piecewise polynomial function that interpolates a series of points in three dimensional space known as the guiding polygon. A cubic polynomial is usually used as this is the simplest function that can represent a change of sign in the curvature. It also has the advantage that conic curves (a common instance) can be represented *exactly*. If a single point in the guiding polygon is altered, the B–spline function is only altered in the local neighbourhood of that point. Variation in the B–spline function is always less than in the guiding polygon. A three dimensional patch (a Coons surface patch) is specified by using B–splines to define a perimeter. Blending functions (often B–splines) can be used to interpolate the surface across the patch. Coons patches can be joined into larger configurations by using common B–splines along the perimeters of the patches. If necessary a join between two patches can be made smooth by ensuring that the first and second derivatives of the blending functions at the patch boundaries are equal.

While this technique can produce very realistic representations of almost any object its applicability to computer recognition is limited. The points of the guiding polygon that make up the model do not correspond to the features of the object and, except in very simple objects, are not recoverable from the image. Before recognition the appearance of the model features would have to be rendered by ray–casting or some similar technique that is computationally expensive.

Nonetheless, Hanson and Riseman [Hanson & Riseman, 1978] having realised the shortcomings of the generalised cylinder primitive chose to use B–splines and Coons surface patches in their VISIONS scene interpretation system. The basic assumption in their modelling scheme was that most objects were not symmetric in all dimensions and had certain natural axes by which they could be orientated. As in ACRONYM, volumes were represented by sweeping either predefined shapes or B–spline enclosed shapes along an axis. The surfaces of an object were represented by positioning Coons patches, each with their own coordinate system in the coordinate scheme of the object around its natural axis.

Fisher [Fisher, 1986] used surface patches as the modelling primitives in his IMAGINE recognition system. Surfaces were either planar, cylindrical or toroidal and their perimeters specified by elliptical or straight line segments.

This modelling system was extended to become the Suggestive Modelling System (SMS) [Fisher, 1985] for use with the IMAGINE–II [Fisher, 1989a] recognition system. The primitives used here were a combination of edges, surfaces and volumes and were used to construct, as the name implies, a model that was *suggestive* of the original rather than an exact replica. An object part was represented by a configuration of both surface and volumetric primitives known as a "primary assembly". Configurations of primary assemblies formed larger assemblies called "structured assemblies". Structured assemblies in turn were used to form the model. Each assembly was defined within its own coordinate frame. By specifying the geometric transformation between different assemblies and primitives, important relations between an object's parts were represented.

To represent object variation, the model parameters such as the curvature of a surface or the position of an assembly were represented either by constants or by variables. Variables were defined within different levels of assembly to reflect their scope within the object. For instance, the joint angle between two robot fingers was defined at the finger assembly level while the scale variable was defined for the hand assembly as a whole. The values that a variable could take could either be unconstrained such as the orientation of a freely rotating robot wrist joint or constrained to fall within certain numerical limits such as the acceptable limits of the scale of the robot hand.

This modelling system is hypothetically superior to those described previously but as yet the object variation facilities have not been used in the recognition process.

In his work on surface reconstruction Terzopoulos [Terzopoulos, 1983] represented a three dimensional object's surface by the use of a deformable "thin plate" model. It was assumed that the object's surface had been sampled over a (possibly large) number of points at each of which the position or depth of the surface was known. These sampled values could be thought of as being represented by pins whose length was determined by the depth. The thin plate model was attached to these pins by the use of ideal springs and allowed to deform to an equilibrium state in which the potential energy of the springs and the thin plate was at a minimum. By dividing the plate into successively smaller grids, each area enclosed by the grid containing a single spring, a hierarchical coarse to fine reconstruction was achieved both to represent the object's surface to different levels of accuracy and as an aid to efficiency.

Terzopoulos [Terzopoulos *et al*, 1987b] [Terzopoulos *et al*, 1987a] approached the problem of three dimensional shape representation with the use of deformable bodies. The deformable bodies could be considered as objects composed of an abstract elastic material. A deformable spine ran through the middle of the construct to give the deformations an affinity for three dimensional symmetry. Using a monocular image, the "forces" acting upon the imaged object were estimated and used to distort the deformable body. The body was deformed until its two dimensional projection on the image plane was consistent with the two dimensional silhouette of the object being modelled.

Both these representations were not intended as models with which to recognise an object but rather a means of representing an object or object surface as it appeared in the image.

2.1.4 Summary

Modelling three dimensional objects with volumetric primitives has the advantage that the primitives are able to represent the three dimensional aspects of the object simply and efficiently. Unfortunately, representation for recognition concerns not only the ease with which an object can be represented but how accessible the model data is when it is to be matched with the type of data available in the image. Volumetric primitives are poor in this respect, their appearance or projection in the image having to be calculated before matching can take place. In this thesis therefore, surfaces are used for object representation since this enables both simplicity in the representation of objects and the ability to match the surface primitives directly with the type of image data used in this thesis. The model structure chosen borrows heavily from SMS in that a hierarchical arrangement is chosen, clusters of surfaces forming parts and parts forming larger parts and, eventually, the model itself.

The structure of SMS is reflected in the model parameters used in this thesis. Instead of using a wide range of parameters to represent the different properties of different types of object, a structured arrangement of just three types of parameter are used whatever the object being modelled. The use of these parameters is fully explained in later chapters.

2.2 Object Recognition

Brooks [Brooks, 1981] addressed the problem of parameterised three dimensional object recognition in two dimensional scenes with the ACRONYM system. The image data was processed to yield features that were projections of the lengths (ribbons) or ends (ellipses) of the generalised cone primitives described earlier. These primitives were matched with the corresponding models using a non–linear Constraint Manipulation System (CMS) that used SUP/INF methods to represent and propagate the bounds of the values of parameters within the constraints. As image features were interpreted, the bounded constraints were passed upwards. When making predictions of image features, a prediction–graph was used and the constraints were passed downwards.

Ribbons were grouped into clusters for which the accumulated constraints had to be consistent with the hypothesised model.

The local feature focus method [Bolles & Cain, 1982] was developed to address the problem of two dimensional object recognition. The objects to be recognised were two dimensional objects lying on a flat surface. The image that formed the input to the problem contained an unknown number of objects of unknown identity that might be only partially visible.

For each type of object to be recognised, a model was constructed that contained not only a list of its features but a strategy for recognition. This strategy identified particular features of the object as *focus features*. For each focus feature, an ordered list of further object features that were likely to be identified in the vicinity of the focus feature was constructed.

Recognition commenced with features in the image being identified. Each feature identified was matched with the list of focus features. If a correspondence was achieved then the strategy associated with that feature *focussed* the search to those features listed as associated with the focus feature. For those features for which correspondences were found a graph of correspondences was constructed. The next stage in the recognition process was to take the graphs produced and to identify the largest consistent subgraphs using a maximal clique method. The consistency of a subgraph was established according to the following constraints:

1. Two model features must not match the same object feature.

2. Two object features must not match the same model feature.

3. Two object features must refer to features in the same model.

4. The distance between two object features must be similar to the distance between the two corresponding model features.

5. The orientation between two object features must be similar to the orientation between the two corresponding model features.

The largest consistent subgraph was that used to form a hypothesis of the object's identity and position but it was possible, where there were more than one subgraphs of the same size, that more than one hypothesis would be generated.

Each hypothesis was then verified by searching the image for any features represented in the model for which correspondences had not yet been found. If new correspondences were found then the estimates of the object's position and orientation were updated. The boundary of the object as represented in the model was projected into the image and the degree of fit between the projected boundary and that of the object established. If the fit was sufficiently good and a certain proportion of features had been identified then the recognition process terminated returning the identity and location of the object. Where relatively few features could be identified it was possible for a match to be ambiguous and it was marked as such.

Faugeras and Hebert [Faugeras & Hebert, 1983] proposed an algorithm that not only matched object and model features but estimated the position of the object in the camera coordinate frame by finding an optimal transformation between the model coordinate frame and that of the viewer. Both object and model were described in an identical manner in terms of planar surface regions. The object surfaces were specified in terms of a viewer centred coordinate frame while the model surfaces were defined in terms of the model's own coordinate frame. If both object and model were rigid then a match existed if and only if there was a transformation that would *exactly* map one to the other. A match consisting of a list of object to model correspondences (possibly incomplete) and the rotation and translation of the object in the image was achieved with a simple tree search procedure. Each image region was taken in turn and an acceptable correspondence searched for amongst the remaining unmatched model regions. Only one correspondence for each region was allowed. The process of identifying an acceptable pairing was carried out in two stages. Initially the dot product between the candidate object region and an object region already matched was compared with the dot product between the corresponding model regions as a fast local consistency check. The use of a fast consistency check to eliminate those pairings that were strongly inconsistent produced substantial improvements in the speed of the search. For the remaining candidate regions that passed this check, a more rigorous and computationally expensive estimate of the error associated with the match was estimated. Normally two object to model correspondences were required to establish the rotation and three correspondences to estimate the translation. To avoid traversing the entire search tree until three correspondences had been achieved, initially only the rotation was used in the error calculation, with the full estimate of the error involving the translation being reserved for the best solutions towards the bottom of the search tree. As a pairing was accepted and added to the list of matches, a least–squares fit was used to estimate the optimal rotation and translation based on the correspondences achieved so far. A further improvement in the efficiency of the search was achieved by storing the best estimate parameters at each level in the tree so that should no match be found and backtracking occur there was no need for them to be recalculated.

A similar approach was used by Bolles and Horaud in the extension of the local feature focus method to three dimensions in the 3DPO system [Bolles & Horaud, 1986]. The initial stage was to look for key features, called focus features, in the image. Which features were regarded as focus features took into account the uniqueness of the feature, its contribution if identified and the likelihood and cost of detection. The choice of a second or third focus feature was even more complex as it had to take into account those features that had already been detected. For example the detection of certain object features might indicate that other object features were self–occluded and should not be considered. This "grow and match" approach had several purposes. Newly identified features could reduce the number of interpretations, verify an existing interpretation, determine a previously unknown degree of freedom or increase the precision with which a degree of freedom was known.

The 3DPO system used several different types of data as its input. Depending upon the type of focus feature, the next features to be searched for were those that would unambiguously identify the object. Further features were selected in order to estimate the position and orientation of the object.

Verification of the generated hypotheses was achieved by using them to predict image features and to compare these with the scene. If too many predictions disagreed with the image data the hypothesis was rejected. Rather than use the hypotheses to predict object features 3DPO used them to make predictions about the object at the data level.

Goad's work [Goad, 1985] concentrated on computer recognition in its strictest form. Here the exact shape of the object was already known, only its position and orientation relative to the viewer remained to be resolved. The computation required was split into two sections. Firstly there was the analysis, or precomputation stage, dealing with unchanging information about the object and secondly there was the runtime stage using the precomputed information for rapid recognition.

Starting with the description of the model a simple depth first search was used to match the features of the model with the features present in the image. The matching process involved three stages:

- Using the current hypothesis of the object's position, the next unmatched model feature was taken and its position and orientation in the image predicted.

- Search the image for possible candidates.

- If there is a successful match then refine the hypothesis using the new data.

If a match was obtained that indicated the presence of an undetected image feature then a search was made of the image for that feature. If a match was found to be inconsistent then backtracking occurred and an alternative match was tested.

A depth first search could be applied to any model/object combination. However, for a particular object certain matches would never occur; hence there were nodes in the search tree that need not be evaluated. To increase efficiency the general search tree was "unwound" and those branches inappropriate for a specific model and pose were pruned. The search tree produced was specific to a particular model rather than generally applicable so it was stored as part of the model. Carrying out this work at "compile time" and placing the results in the model meant that the computation required to search the tree at "run–time" was greatly reduced. As a further aid to efficiency, the geometric calculations required to compute the relative positions of an object's subparts were also carried out at compile time and the results placed in a look–up table again stored within the model.

A solution to the problem of identifying an object from amongst a known set and locating it relative to the sensor was proposed by Gaston and Lozano–Pérez [Gaston & Lozano-Pérez, 1984]. The object to be identified was a single polyhedral object placed on a flat supporting surface. A number of points (normally three) on the surface of the object and surface normals at those points were sensed a known height above the plane supporting the object. This reduced the matching problem to that of two dimensions with one rotational and two translational degrees of freedom.

The object was sampled at f points P_i that were known to be on the surfaces of n models O_j each having e_j edges. The range of possible pairings were represented by an *interpretation tree* (see Figure (2–1)). One tree was required for each model. Each node in the tree had e_j descendants corresponding to the different interpretations of a sensed point. There were f levels in the tree, each level representing the interpretations of a given sensed point. Sensed object points were allowed to belong to the same model edge so the number of branches at each node remained the same.

Because the interpretation tree represented all possible interpretations then the correct interpretations (there may be more than one, especially in a symmetric model) would always be represented.

However, the search space was vast with the total number of possible interpretations given by:

$$\sum_{j=1}^{n} {e_j}^f \tag{2.1}$$

To reduce the number of interpretations the solution proposed was to identify inconsistent sensed object to model pairings and to remove the corresponding node from the interpretation tree. Since this meant the removal of the subtrees beneath that node, the saving was significant. Obviously, the higher the level within the tree that this could be achieved, the greater the saving.

Three constraints were developed to identify inconsistent pairings:

1. The distance constraint – The distance between a pair of sensed points must be a possible distance between the model edges paired with them in the interpretation.

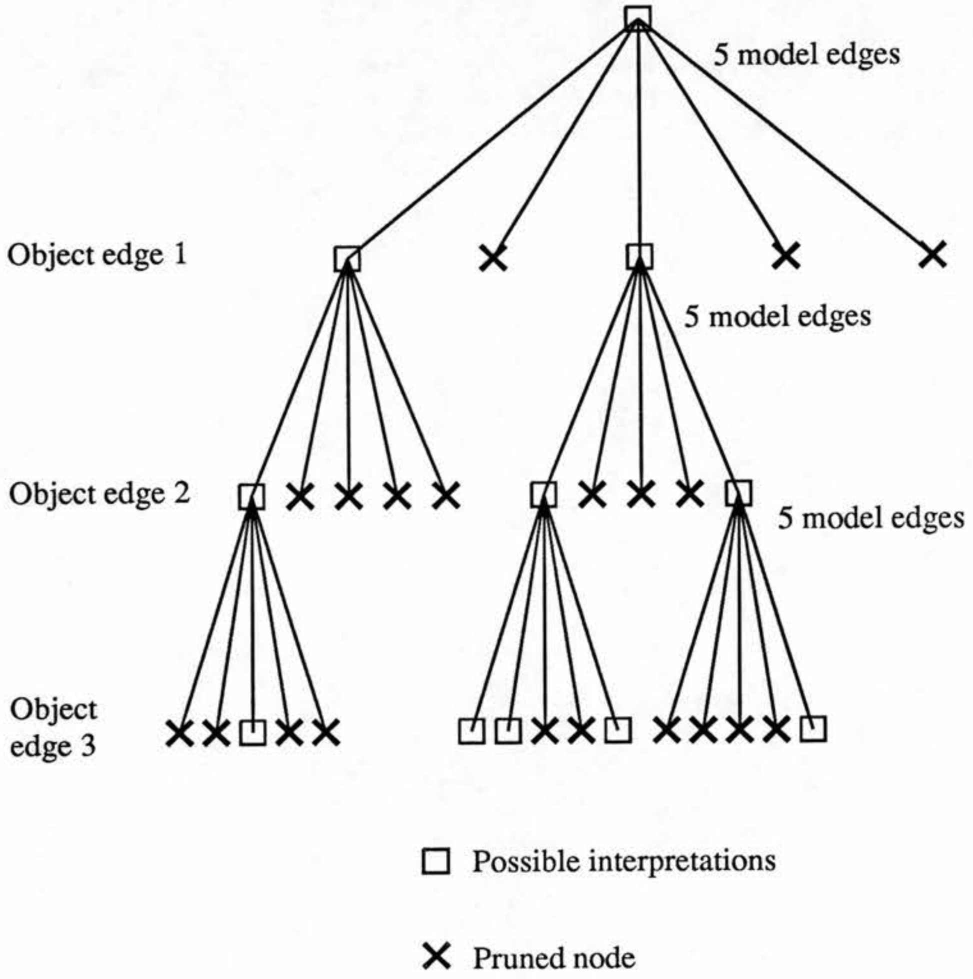


Figure 2–1: The Interpretation Tree Used by Gaston and Lozano–Pérez

2. The angle constraint – the range of angles between a pair of sensed surface normals must include the angle between the model normals paired in the interpretation

3. The model constraint – the positions of the sensed points must satisfy the equations of the edges paired with them for some orientation and translation of the object.

Of the three constraints the first two were the most efficient. They required pairings between only two sets of sensed points and model edges so that pruning could take place at the second level of the tree. The data required to determine consistency with the constraints was available locally. The last constraint required three pairings that meant that pruning could only be carried out at the third level of the tree. Furthermore the transformation that mapped object to model had to be determined, a computationally expensive operation.

In the examples tested only three object points were sampled. This often led to ambiguous interpretations. Several enhancements to overcome this were proposed. One of these used the acquisition of a second set of sensed points chosen deliberately to reduce any ambiguities. A further enhancement proposed was the use of a hierarchical structure for particularly large models.

Grimson [Grimson & Lozano-Pérez, 1984] extended this work further to work with polyhedra (modelled by surfaces rather than edges) with six degrees of freedom. Two new constraints were proposed:

1. The direction constraint – The range of values of the component of the vector between two sensed points in the direction of the sensed normals must be within the range of components for points on the model surfaces paired with them.

2. The triple product constraint – The sign of the triple product of three sensed normals must be the same as the sign of the triple product of the normals to the three paired faces.

Other constraints such as the area enclosed by three points and the visibility of paired surfaces were proposed. These simple geometric constraints used data that was locally available between the sensed points and the model. This had the advantage that the constraints could be implemented by using lookup tables reducing the computational effort.

The efficiency was further increased by ordering pairs of sensed points on the basis of their separation, the more distant points being ordered first. This tended to ensure that the distance constraint was able to operate most efficiently at the highest levels of the tree.

Taking the distance constraint as an example it was shown that this limited the upper bound on the number of interpretations to a linear function of n for two dimensional objects and to $n^{(k+1)/2}$ for three dimensional objects, where n was the number of model surfaces and k was the number of object surfaces.

The function of an interpretation tree search was to find a consistent match of *all* the features in the image with the features represented by the model associated with it. For this to work efficiently all the image features must be from the same object. Any spurious features or features from a different object will not be represented in the model and a consistent interpretation is not possible.

Grimson [Grimson & Lozano-Pérez, 1987] showed that the interpretation tree search could be extended to identify polyhedral objects represented by planar surfaces in an environment containing overlapping parts. The basic interpretation tree was extended by the addition of a "null face" below each node in the interpretation tree. At each level of the tree an object surface could be tested for consistency with each of the model surfaces or if found to be inconsistent could be rejected from the interpretation by assigning it to the null face. The representation of extraneous data in this way worked well but the number of interpretation generated was very large. If, at some level of the interpretation tree, an interpretation was valid, then all subsets of interpretations would also be valid since all assignments of the null face are legal.

To reduce the number of interpretations to an acceptable level a variety of approaches were tried with varying degrees of success. The first approach was *heuristic* *search with cut off*. Since each valid interpretation had a large number of valid subsets a heuristic was introduced to attempt to find the "best" interpretation. The metric chosen for the heuristic was the combined area of the object surfaces in the interpretation. An external variable was used to track the best interpretation as the interpretation tree was searched in a depth–first manner. For each node in the tree the combined area of the object surfaces assigned to non–null model surfaces was recorded. Also recorded was the combined area of all the object surfaces further down the tree still to be assigned. Obviously an interpretation was only worth continuing if the area of the assigned surfaces and the area of the surfaces still to be assigned was greater than the largest combined area found so far. If a complete interpretation that used no null–face assignments was found then no interpretations that included null–faces were accepted. It was discovered that the correct interpretation was determined fairly early on in the process but that large amounts of computational effort were expended trying to improve upon it. This problem was overcome by *cutting off* the search when the combined area of the object surfaces in the interpretation reached some percentage of the area of the model. The use of a heuristic to identify good interpretations and the use of a cut off point to limit there expansion provided a great improvement in efficiency.

A further method that was tried was that of coupled constraints. The constraints used previously were used only to test the validity of a single object to model match. There had been no attempt to link the constraints to check that the matches in an interpretation formed a consistent whole. The method chosen was to use constraints to compute a set of ranges for possible object to model matches. As constraints were applied further down the tree these ranges could be reduced. If at any time no consistent range could be found then the interpretation was regarded as inconsistent. This approach required a high degree of computational effort but returned only small improvements in performance.

Finally a Hough Transform was used to attempt to cluster surfaces that might belong to the same object into "buckets". An interpretation tree search would then be used to find any consistent interpretations of the surfaces within a bucket. This technique also provided an improvement in efficiency but less than was hoped since of the surfaces contained in a bucket up to a third could be extraneous data [Grimson & Huttenlocher, 1988] and an exponential search would result.

The most efficient approach was a combination of two of these techniques. The Hough Transform was used to divide the image data into subsets and for each a heuristic search with cut off was used to find consistent interpretations.

Where the features in an image are known to have originated from the same object then the size of the search space is a quadratic function of the number of model features. If, however, the image contains clutter, or features from other objects, then the size of the search space quickly becomes exponential. It was shown [Grimson, 1988], [Grimson, 1989] and [Grimson, 1990b] that terminating the interpretation tree search using a heuristically determined cut off threshold could reduce the search space to a polynomial function of the number of features for an image with small amounts of clutter. For large amounts of clutter the search again became exponential and it was shown to be necessary to identify those subsets of image features belonging to the same object before efficient matching could take place.

This work was extended [Grimson, 1987] to handle the recognition and localisation of two dimensional object classes or families. The model parameters used enabled object families of three types to be represented. An object could show a uniform change in scale. It could contain a number of subparts that could rotate or the object could appear stretched along an axis. Representation of possible object to model pairings was achieved using the interpretation tree described previously. Because of the variations exhibited by the different classes of object the constraints developed previously to eliminate incorrect interpretations could no longer be used. New constraints were developed that involved the use of a scale, stretch or rotation parameter. Because the constraint parameter could take a wide range of values for a single pairing, the constraints were coupled. In this way the range of values that the parameter could take was narrowed as further pairings were made. If the set of possible values became empty then the interpretation was regarded as inconsistent and pruned. For each of the interpretations regarded as valid a value, or possibly a range of values for the parameter would be returned.

The use of null nodes to represent spurious data in an interpretation tree was used to address the problem of mobile robot localisation [Drumheller, 1987]. The input to the problem were a number of edges obtained by sonar rangefinder corresponding to walls of the room in which the robot was positioned. The basic recognition technique was the same as that described earlier using an interpretation tree to represent possible image to model pairings and a null-node to represent spurious data. To prune the number of inconsistent interpretations the standard constraints of distance, angle and normal direction were used. Each remaining interpretation was tested for global consistency by attempting to establish a position and orientation of the robot consistent with the range data received.

Once the position and orientation of the robot associated with each interpretation had been determined then the number of interpretations could be reduced further by the application of a further constraint – the sonar barrier test. The original range samples obtained from the sonar were projected back to establish those that intersected the hypothesised positions of the walls. Those that did had to return a range value that was inside the wall or outside the wall by no more than a small margin otherwise the interpretation was rejected. The efficiency of the process was increased by only considering those walls whose orientation to the robot was such that the sonar system could have received a return echo.

The HANDEY project [Lozano-Pérez *et al*, 1987] was developed to locate an object in a range data image and to move it to a different location. Objects in the image were represented by three dimensional edge segments and were expected to have a corresponding model also constructed from three dimensional edge segments. Although there might be many models the object to be recognised was identified in the MOVE TO command reducing the problem to that of a known object but unknown orientation. Object localisation was achieved by matching object and model edges with a hypothesis and verify algorithm. The interpretation tree used for this purpose was a variation on the original method described in that a pair of edges from the model were matched with all possible pairs of edges in the image. Each match was sufficient to determine up to four transformations that mapped the model coordinate frame to that of the object. The ambiguity in estimating the transformation arose due to different possibilities in assigning direction vectors to the line pairs. Once the transformation was known, this could be used to predict the position of further edges. The match that predicted the greatest number of edges was the one chosen.

To reduce the number of interpretations generated two constraints were developed:

1. Edge length – The length of an object edge must be less than or equal to the matched edge in the model.

2. Relative pose – The relative pose of two edges was defined as a function of the distance and angle between them. The pose of a pair of object edges must be consistent with the pose of the corresponding model edges

The edge length constraint was particularly efficient in that it needed a match between only one object and one model surface. The relative pose constraint needed a match between a pair of object and model edges.

A similar approach was used in the recognition strategy using object alignment proposed by [Huttenlocher & Ullman, 1987]. Both the objects and models were flat and rigid (two dimensional). The objects were positioned in three dimensional space introducing the problem of objects appearing distorted in the image, for example by foreshortening. Recognition was achieved in two stages. In the first stage a minimal number of *alignment points*, normally three, were matched and the corresponding transformation, consisting of a rotation, translation and a scaling operation, between the object and the model (if a consistent transformation existed) determined for each.

The number of possible alignments was large. If there were m model features and o object features then the number of alignment point triplets were $(im)^3$. Not all the triplets would yield valid transformations. For those that did, the calculated transformation was used to project the model into the image and the transformation scored by how many model features were mapped onto the corresponding object features.

An extension was proposed for objects that were three dimensional. Once the alignment had been calculated, those model features that were hidden (self–occluded) were removed before the model was projected into the image. A check was made to ensure that the three points used did not correspond to any of the hidden features. An alternative approach was suggested where rather than using a single three dimensional model a large number of planar views of the object taken from different viewpoints were used. The alignment between each of these and the object in the scene could be determined essentially reducing the problem to that of two dimensions.

Ikeuchi [Ikeuchi, 1987] used an interpretation tree pre–generated from a CAD model to localise a three dimensional object of known identity. The image was a range data image that could contain more than one object. The upper branches of the interpretation tree represented the existence in the image of a surface corresponding to an individual surface of the model. If the surface was visible then that node in the tree was expanded, if the surface was not visible then it was pruned. This was repeated as necessary at nodes further down the tree, each sequence of nodes representing a list of object surfaces that were visible. The list of visible surfaces enabled the attitude of the object to be classified uniquely as belonging one of a number of "attitude groups". These attitude groups were further subdivided into different attitudes, each attitude represented by a different leaf node in the interpretation tree and giving the transformation between viewer and object.

Fisher [Fisher, 1986] [Fisher, 1989b] developed a data driven system that used image data to *invoke* models from a model database. Invocation took place within image contexts examples of which might be a surface hypothesis or an assembly of surfaces. The basic structure of invocation was a model instance. A model instance could be a model, a model part, a configuration of features, a feature, a surface or an edge. The invocation of a model instance as an interpretation of the image data within an image context was based on suggestion. Suggestion came about by either evidence or association. Evidence came in two forms, the direct evidence of image features or the indirect evidence given by associated hypotheses. As an example, a toroidal shape would give direct evidence towards the existence of a bicycle wheel while the existence of a bicycle frame would be only indirect evidence towards the existence of a wheel.

The evidence was accumulated into a measure of the plausibility with which a model explained the image structure. As new pieces of data became available these were integrated into the plausibility values. Invoked models were used to form hypotheses that were then verified. If the verification was successful then the plausibility of the verified object was permanently set to one. This could provide direct evidence towards further invocations either by association or by direct evidence within a larger image context.

Associations provided indirect evidence and were divided into four types. An object could be a more general class (supertype) than another. An object could be a more specific class (subtype) of another. An assembly could have another object as a component (supercomponent). An object could be a subcomponent of an assembly (subcomponent). A high plausibility of a structure could also be used to inhibit alternative interpretation.

The hypotheses constructed by the invocation network were represented as nodes in a graph. These nodes were linked by direct and indirect evidence relationships.

The constructed hypotheses were then passed to the hypothesis completion process for verification. Verification was achieved by estimating the coordinate frame for the invoked structures, predicting which of those model features represented were not visible due to self–occlusion and finally finding the other model features or explaining their absence.

Later work [Fisher & Orr, 1987] extended this approach to incorporate the use of a SUP/INF geometric constraint network to propagate constraints for a consistent match.

A strategy for data driven model selection was also proposed in [Knapman, 1987]. Both the objects to be recognised and the corresponding models, stored in a model database, were represented by *feature types*. In the examples given there were two feature types – straight lines and circular arcs. Each feature type had associated with it a number of *properties* such as its length or radius. Between pairs of features were *relationships* between their properties. Tables of these pairwise relationships were constructed for both the objects in the image and the models in the database. The various types of features for all the models in the model database formed vector bases. For each type of property a property vector was constructed, the n^{th} item in the property vector corresponding to the n^{th} item in the vector base. A given property or relationship in the scene could be matched with those in the model database and a vector of Boolean values produced. The size factor between objects in the scene and the models in the database was unknown and so, for scale dependent matches, there was a score vector that not only contained the Boolean vector but also a vector containing pairs of size factors representing the upper and lower limits. Using pairwise relationships the intersections of these score vectors could be found using a logical AND operation on the Boolean vectors. If the ranges of the size vectors overlapped then the new range was the intersection. If they did not overlap the score vector was set to zero. To combine the score vectors, offsets were used to establish relationships between the property in one vector and the property in another. By finding the intersections the ranges of the size vectors could be further refined.

The strategy devised for data driven model selection was to choose a pair of features from the scene and to compute the score vectors. If the number of models that matched the pair of object features was too large then a third object feature was chosen and its score vector computed. This process continued until either:

1. The number of models identified was less than a certain threshold.

2. The number of object features chosen exceeded some limit.

3. An object feature was chosen that excluded all the remaining models. (This occurred where there was more than one object in the scene, or when noise or bad data was chosen). When this occurred the process backtracked and the last scene feature chosen was removed.

The technique of generating and then testing possible interpretations of sensed data was also used by Ayache and Faugeras [Ayache & Faugeras, 1986] in their HYPER system. The problem approached here was to identify and locate one or more randomly orientated, partially occluded, flat industrial castings in an *image* of unknown scale. The scale of the image affected the *size* of the objects as represented by the image. Within a given image, all the objects had the same scale.

The first stage in the generation of an interpretation or hypothesis was to pair a "privileged" model feature (an unusually long edge) with a compatible feature in the image. Edge segments were used in both the model and the image but the technique is applicable to other primitives. Compatibility was established by using local, intrinsic features such as the angle between a model feature and its nearest neighbour and the angle between the image feature and its nearest neighbour. If the image scale was known *a priori* then the ratio of distances between the model feature and its nearest neighbour and the scaled distance between the image feature and its nearest neighbour could also be used. Where the scale was unknown it was estimated from the ratio of the model and image edge segment lengths. The angle of orientation of the object was estimated by the angle between the model and image edge segments. From this an initial estimate of the translation parameters could be made. As a first estimate of the objects position the estimated values for the scale, rotation and translation were usually inaccurate. A error covariance matrix was used both as an estimate of the error and to rank the compatibility of the hypotheses generated, generally some several hundred in number.

The best ranked hypotheses (normally the top few tens) were then tested. With an initial estimate of the scale of the image and the rotation and translation of the object in the scene available, the positions of any unpaired model features could be predicted. This was done iteratively with the unpaired model feature closest to the privileged feature being transformed by the current estimate of scale, rotation and translation to predict the position of the corresponding object feature in the image. By using the model feature closest to that already paired, the effect of any errors in the current estimate of the transformation were minimised. To determine which of the image features was best paired with that predicted from the model a dissimilarity function was used. The dissimilarity function used a weighted sum of the difference between the predicted feature and each feature in the image. The difference took into account differences in the orientation, the length of and the distance between edge segments. The model feature was paired with the image feature that returned the least dissimilarity or if the dissimilarity function failed to return a value beneath some threshold then the model feature was deemed to have no pairing in the current hypothesis. As new pairings were obtained between the model features and those of the object the estimate for the scale of the image and the position of the object were updated using a recursive least–squares (Kalman filter) technique. Testing of the hypotheses terminated when all the hypotheses had been tested or a very high quality of fit had been calculated.

An interesting combination of the interpretation tree approach and the local feature focus approach was used in the TINA project [Pollard *et al*, 1987]. This project aimed to match different views of three dimensional scenes but the techniques used are equally applicable to the matching of a three dimensional model with a scene. Different scenes, or the model and scene were both represented by three dimensional edge segments.

The exhaustive tree search was replaced with a hypothesise and test approach based on a number of "focus features". Unlike previous work using feature focus nothing nothing was know in advance about the scene and the focus features and matching strategies had to be generated at run time. The salient features used to focus initial attention were particularly long edges in the image. Some pre–processing was done to ensure that a suitable edge was identified for all sections of the image.

The matching process proceeded as follows:

1. Each focus feature was taken in turn and the S significant features closest to it identified. (The significance of an edge feature was determined by its length.)

2. The focus feature was *conservatively* matched against the features of the model, again on the basis of length. Given a plausible match consistent matches for each of the neighbouring S features were found.

3. Each set of matches was searched for a maximally consistent clique of size C.

4. A cheap tree search was used to extend the clique by identifying new matches for other edges in the image that were consistent with the matches in the clique.

5. Finally the extended cliques were ranked on the basis of the amount of image data they explained, the total length of the matched edges.

Each focus feature could be considered as the origin of an independent search tree. Since each focus feature had the potential to determine the same clique of matches, allowing only conservative matching of the focus feature itself did not hinder the matching process. Redundancy due to consistent cliques obtained from different focus features was avoided if they could be identified and combined during the search for further matches.

If the number of features n in a scene were to increase then so too would the number of potential matches. The number of transformations and the cost of each transformation would also increase. However, regardless of the number of features in the scene S and C would remain the same and so the cost of exploring each focus feature remained constant though there would be more focus features to explore.

Murray [Murray, 1987b] [Murray, 1987a] applied an interpretation tree search to the problem of recognising three dimensional objects in motion. The three dimensional information obtained from visual motion is subject to the depth:speed scaling ambiguity that means absolute depth and size information is unavailable. If absolute depth and size data were available, for example from an active rangefinder, then the problem could be considered as recognising objects of different sizes or scales. Both objects and models were described by planar surfaces whose scaled position and surface normals were known.

Without absolute distance information the distance constraint developed by Grimson could no longer be used. Instead five new constraints, that were essentially scale invariant, were developed. Four used the surface normal vectors and the direction vector between *pairs* of object and model surfaces to determine a consistent match. The last constraint developed was a modified distance constraint using matches between *triples* of object and model surfaces. Although the distance between any two object surfaces differed from the corresponding model distance by an unknown scale, the ratio of the two distances between three surfaces should be the same between object and model.

Once a valid interpretation had been identified it was validated by finding a globally consistent transformation between model and object. This involved finding a consistent rotation, scaling factor S and translation by a least–squares fit.

In later work [Murray & Cook, 1988] an interpretation tree was used to to match fragmentary three dimensional edge data with models also represented by three dimensional edges. Model edges were defined by their end points or terminators and by arbitrarily defining one as a start terminator and the other as the end terminator the edge segment could be given a direction either positive or negative. The directions of the model lines were arbitrary.

The image data composed of three dimensional straight line fragments that were assumed to correspond to edges within the model. An interpretation tree was used to represent pairings of edge fragments with model edges. Within an interpretation edge fragments were labelled with a direction. Initially this was assigned as "uncertain" but as the interpretation was formed the direction of an edge fragment could also be labelled as "positive" or "negative" or linked to a further fragment and labelled as having the same or opposite direction to that. To reduce the size of the interpretation tree the constraints used in previous work [Murray, 1987a] were used. The direction constraints were used to *determine* a direction for an edge fragment while the angle constraints were used to *propagate* the directions of edge fragments. The directions assigned to each edge fragment were compared with the directions of the corresponding model edges to check for a consistent match. Where the direction of an edge was determined this was propagated back up the interpretation tree to be used to assign directions to other edge fragments. This could mean that an interpretation in which all assignments of direction up until then had been consistent with the model, would suddenly become inconsistent and the interpretation be rejected. It should be stressed that the directions assigned to edge fragments were only relevant inside a given interpretation. Were the interpretation to be found inconsistent then the tree search would back track and the direction of all edge fragments be reset to "uncertain".

Simsarian [Simsarian *et al*, 1990] addressed the problem of self–localisation of a robot with two positional and one rotational degree of freedom. An interpretation tree was used to match a polygonal map of a room with the walls of the room detected by the robot. The map was divided into View Invariant Regions (VIRs) were only certain walls could be seen. By first identifying those VIRs in which the robot might be positioned, the *width* of the interpretation tree could be reduced since the detected walls need only be matched with those known to be visible in the VIR.

Reid [Reid, 1990] used an interpretation tree in the recognition of parameterised objects. A standard interpretation tree was used to represent all possible pairings of the object features in the image with the features of the model. For each model a constraint propagation network similar to that of Fisher and Orr [Fisher & Orr, 1987] was precompiled. As object to model pairings were generated and the interpretation tree extended, the bounds on the values assigned to the model parameters were refined. If, as a result, the constraints were to become inconsistent then back tracking would occur and that node of the interpretation tree would be pruned. The state of the constraint network was stored at each node in the tree so that if backtracking were to occur then interpretation could continue with the previous bounds. Where a complete interpretation was found then the state of the network at the appropriate leaf node gave the values of the model parameters. These were substituted into a generic model to give a specific instance of the model. The transformation between this model and the object in the image (if one existed) was then determined to establish the object's position.

In recent work [Lowe, 1991] a very different approach has been described. The aim was to recognise and locate a three dimensional object represented by a three dimensional model with any number of internal parameters in a two dimensional image. This was achieved by the use of an iterative process repeatedly updated the estimates of the model's parameters. The predicted model was projected back into the image and any differences or errors between model and object used to aid the next estimate of the parameters.

2.2.1 Summary

The majority of the work described here aims to use the object features represented in the image to suggest the identities of the object, or objects, in the image. The tests, or constraints, applied to object features when trying to determine their identity tend to be rather simple and use data locally available. The object hypotheses constructed in this manner may not form coherent wholes and a verification process using global constraints, such as coordinate frame transformations, is normally used to ensure that a hypothesis is consistent.

The main shortcoming with this approach is that the features in an image can be interpreted in a large number of different ways. If all the features are known to belong to the same object then the use of local constraints can reduce the problem. If however, the image contains one than one object then the application of simple constraints becomes less effective.

Where, as here, an image may contain several unknown objects it is important that the recognition of the objects can be achieved efficiently.

Chapter 3

Representing Deformable Objects

3.1 Introduction

The traditional approach to object representation for computer recognition is to represent each object that the recognition system must identify with a different model. Recognition is achieved when an object is successfully matched with its corresponding model. If just a few, distinct objects are to be recognised this approach works well. However, if a large number of objects are to be recognised then an equally large number of models will be required. Searching a large database for the model corresponding to a particular object is computationally expensive and the larger number of models increases the chance of an incorrect identification. A further disadvantage is that a new object with no corresponding model will not be recognised even if similar to, or in the same class as, other models already in the database.

A better approach is to have just one model to represent an entire class of objects and to use parameters within the model to allow for the differences between individual examples. The number of models is reduced so less computational effort need be expended to identify the correct model and the probability of a correct match is increased. A further advantage is that new, previously unseen, objects, provided that they are members of an existing modelled class, can be recognised without the need to construct a new model. There are however two disadvantages with using parameterised models. Firstly there is the computational expense of determining values for the parameters. Secondly the model parameters must be carefully chosen such that they not only characterise the class and can be used to predict the appearance of an object, but must also be easily obtainable from the object during the recognition process.

The differences shown by members of an object class are many and varied from variable numbers of features to features that simply change shape. Here, the differences between objects in a class are limited to three types. Objects can exhibit changes in size, in all three dimensions, contain parts that show elongation or contraction (stretching) along up to three orthogonal axes and have parts and surfaces that show small perturbations in position and orientation not accounted for by scaling or stretching. The corresponding three dimensional models are constructed in a hierarchical manner, the basic primitives of surfaces forming model parts, configurations of parts forming larger parts known as superparts and configurations of superparts forming still larger superparts or, at the highest level, the model itself.

3.2 Characteristics of Deformation

In this section the three classes of deformation represented in this thesis are introduced. The effects of each on the properties of an object, its parts and its surfaces are described.

3.2.1 Global Scale

The first, and simplest, class of objects to be considered is that where the overall size or scale of an object can change. The scale of an object does *not* reflect its size as depicted in the image but its size relative to the class model. Different objects in the same image may have different scales. The scale of an object is defined as uniformly affecting the object as a whole: each part of the object has exactly the same scale as that of the object. When an object is scaled, the dimensions of the object and the positions of its parts all change by the same factor, but the relative configurations of its superparts and

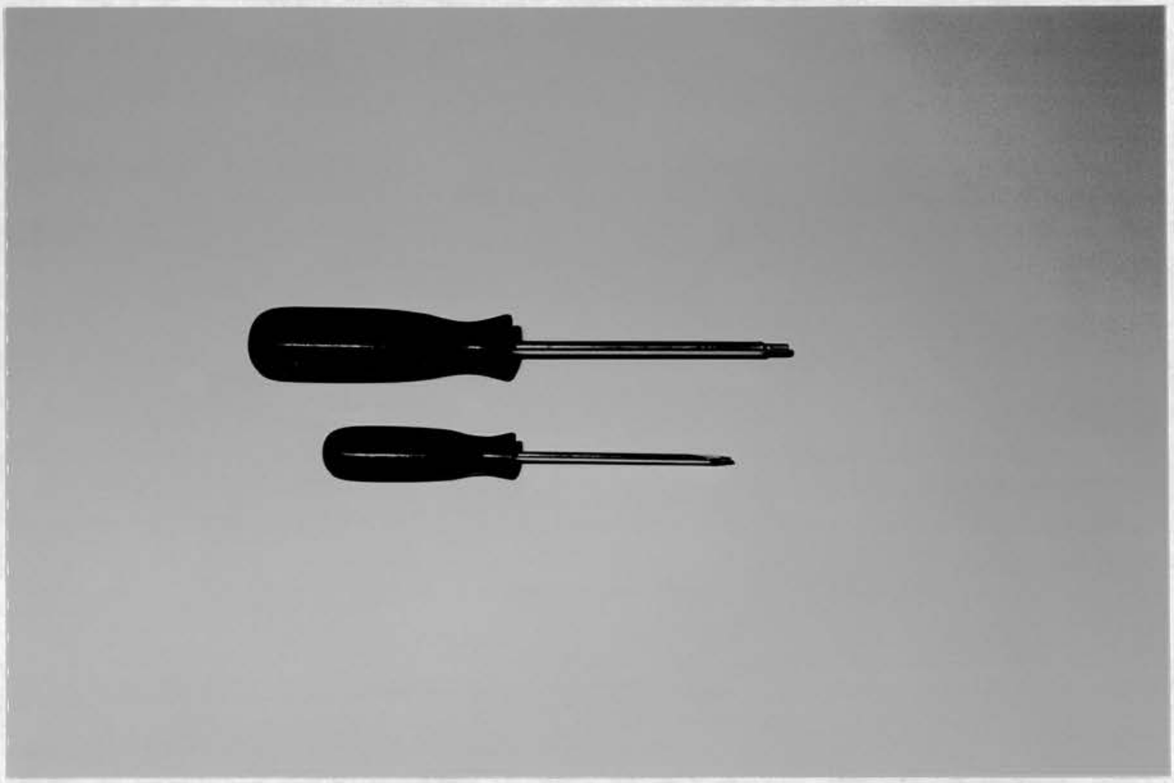


Figure 3–1: A "Standard" Screwdriver and One of Smaller Scale

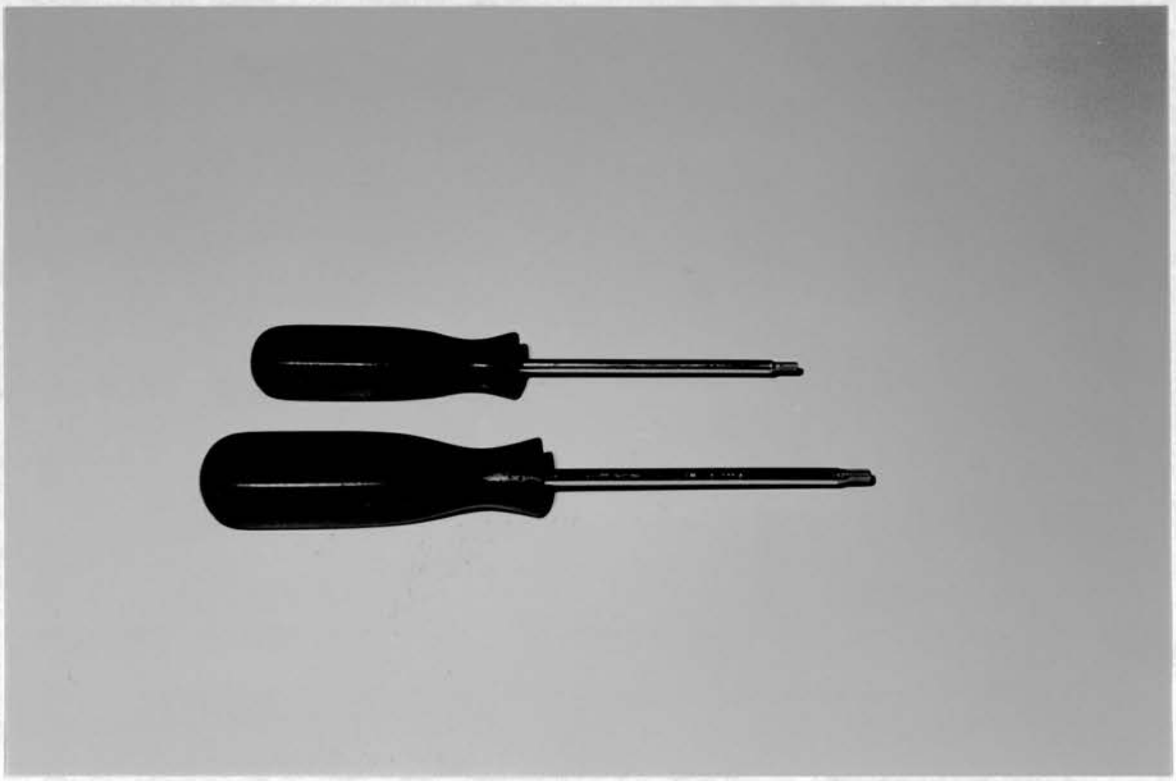


Figure 3–2: A "Standard" Screwdriver and One of Larger Scale

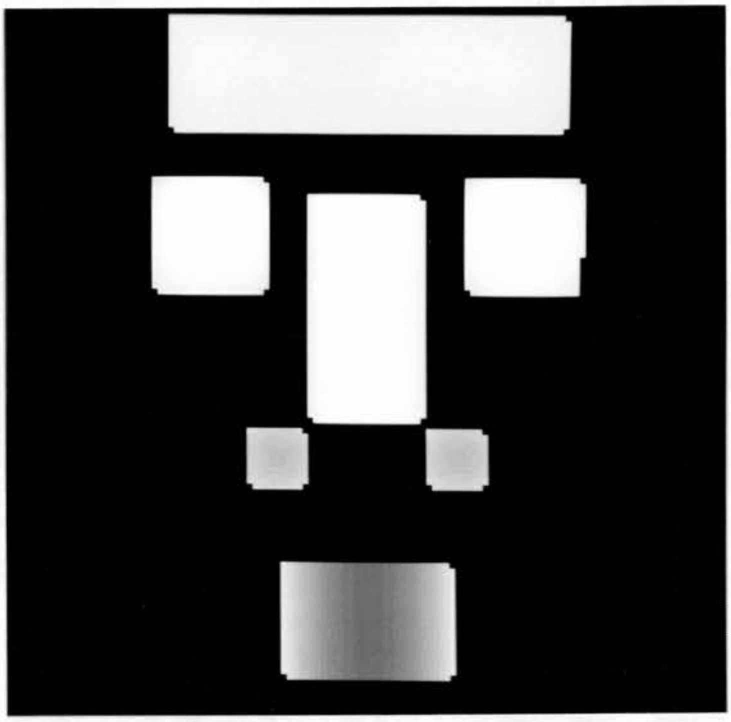


Figure 3–3: The Synthetic Human Face Object Showing No Stretch

parts remain the same. Within each part of the object the position, dimensions and curvatures of each surface also change by the same factor though the configuration and shape (positive elliptical, negative elliptical, hyperbolic, positive cylindrical, negative cylindrical or planar) of the surfaces remain unchanged. If there is a model that is an exact representation of an object that shows this class of deformation then, with an object of arbitrary size, the model can be "scaled" by the use of a constant factor to represent exactly this particular instance of the object. Since this scaling factor affects the model as a whole, it is referred to as the global scale.

Figures 3–1 and 3–2 show an example of a "standard" screwdriver and several instances of screwdrivers of different scales.

3.2.2 Part Stretch

The next class of object deformation to be considered is that where some, or all of the object parts, can show elongation or compression. Since the objects to be

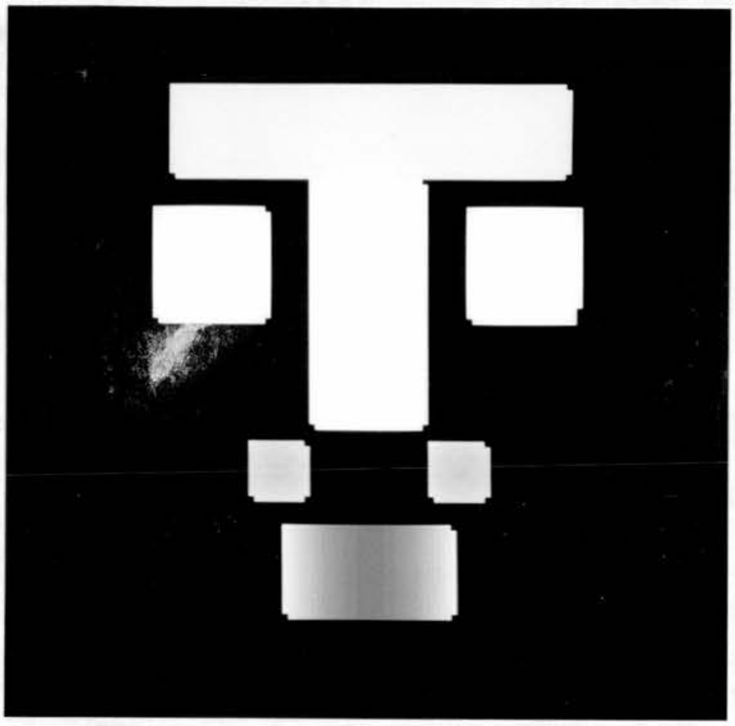


Figure 3–4: The Synthetic Human Face Object Showing an Elongated Nose and Contracted Face

recognised are three dimensional, an object part may exhibit elongation or contraction (more generally referred to as stretch) along up to three independent axes. This form of representation is limited to those objects that exhibit independent stretch in three orthogonal directions. Stretch that is not independent, such as shearing, is not represented, but if it is small then it can be represented by the individual variation discussed in the next section. It should be obvious that an equal amount of stretch along all three axes is equivalent to a scaling operation on a single part. How the effects of scale and stretch are separated is covered in the next chapter.

Unlike the global scale, the stretching of a part within the object and the axes along which it occurs (the stretch axes) are specific only to that part. Different parts of the same object may be stretched by different factors and along different axes. Stretch can be shown by any part of an object including object superparts or the object itself.

The degrees and directions of stretch for a superpart or a model are independent of the stretch factors of its constituent parts, though in practice they are often related. For example, a long, thin nose does not always mean the face will also be long and thin. Similarly a long, thin face might have a short, squat nose. The interrelationship of the coefficients of stretch for parts, superparts and the model itself are discussed in the next chapter.

Where an object has undergone stretching, the positions and configurations of its superparts and parts will also change. In a stretched object part the positions, dimensions and curvature of its surfaces may change though the shapes (as defined previously) of the surfaces remain the same.

If an example of this class of object is exactly represented by a three dimensional model, then an exact representation of an object, or object part that has undergone stretching, can be achieved by transforming the model or model part by a three dimensional stretching transformation. If the coordinate frame of the model, or model part is carefully positioned then the three orthogonal axes of stretch can be made to lie along the axes of the local coordinate frame of the part or model.

Figure 3–3 shows the three dimensional image of a synthetic human face and

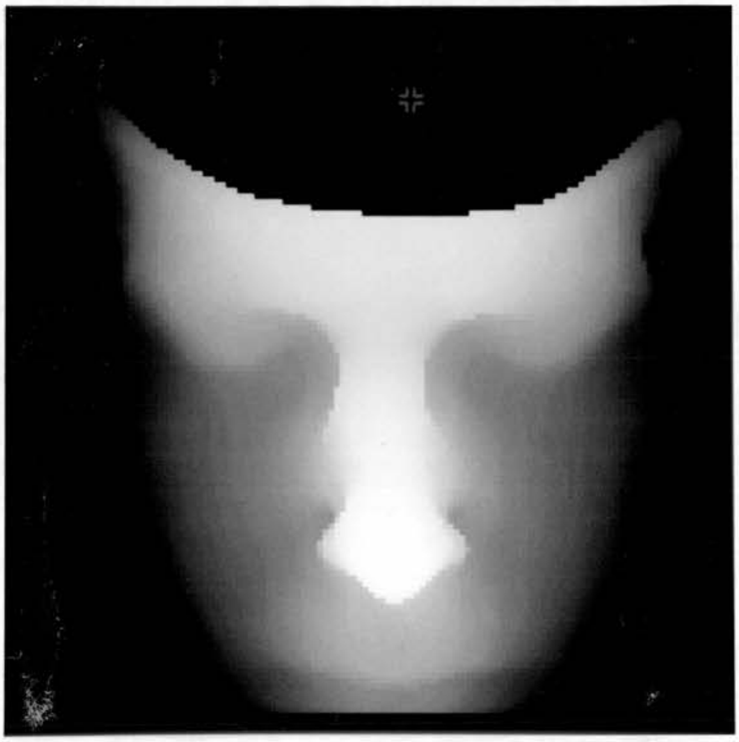


Figure 3–5: A Class of Object Exhibiting Variation

Figure 3–4 shows the same object but with its nose part elongated along its length and the object itself contracted in both width and height.

3.2.3 Individual Variation

Although a given object may have a scale and its individual parts have stretch values, the features of an object or object part will not normally be of exactly the same scale or show the same degree and directions of stretch. Usually an individual feature of an object will show variations in its appearance or position other than those caused by scale or stretch. Where an object's features exhibit variation, then its scale and the stretch of its parts become the "best" approximations for the model or part as a whole.

The individual variations shown by superparts or parts of an object are *small* perturbations in orientation and position, while the surfaces within an object part show perturbations in orientation, position and curvature that cannot be accounted for by the effects of part stretch or global scale. These variations are individual to each part or surface in an object. An exception to this is where an object has two or more identical parts or surfaces, in which case the individual variations exhibited by one may well be present in the other. Small variations in position of all parts along the same axis can be equivalent to a stretching action. How the components of stretch and individual variation are resolved is covered in the next chapter.

Where the surfaces in an object show individual variations, not only do the position and configuration of the surfaces change (though the change is small) but unlike the previous deformations, the shape of the surface may also change if the variation is relatively large or one of the principal curvatures of the surface is close to zero. Unlike the global scale or part stretch parameters the individual variation in the position of a part or surface is *independent* of the variation in the other properties of the part or surface such as its orientation or shape.

Figure 3–5 shows the human nose part and the effects of the individual variations of the surfaces of the object overlaid with those of a model part that has only been subjected to scaling and stretching.

3.3 Object Representation

The modelling system described here is used to construct three dimensional models in a form suitable for object recognition. It is closely related to Fisher's Suggestive Modelling System (SMS) [Fisher, 1985]. Like SMS, the models are constructed in a hierarchical manner using surfaces as the basic building blocks or primitives of the system. Groups of surfaces are configured into model parts – similar to SMS's "primary assemblies". Groups of parts are configured into larger model parts known as superparts – similar to SMS's "structured assemblies". A superpart may consist of both parts and surfaces, but a part consists only of surfaces. Superparts may in turn be part of still larger superparts until at the highest level, a collection of superparts and parts defines the model itself. Figure 3–6 shows a schematic view of a three dimensional model of a screwdriver constructed using this system. This represents the

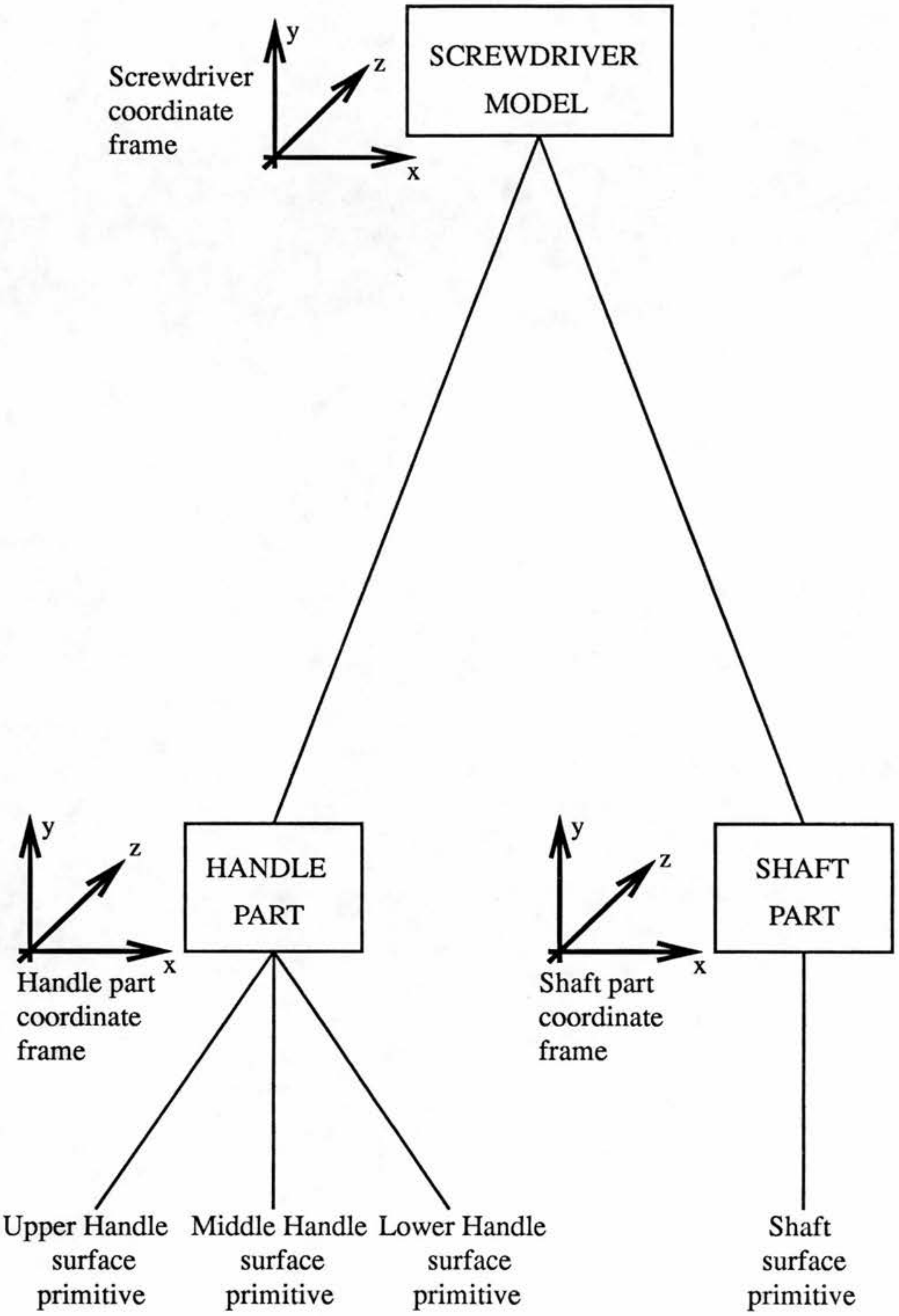

SCREWDRIVER MODEL
Screwdriver coordinate frame
y
z
x
HANDLE PART
Handle part coordinate frame
y
z
x
SHAFT PART
Shaft part coordinate frame
y
z
x
Upper Handle surface primitive
Middle Handle surface primitive
Lower Handle surface primitive
Shaft surface primitive

Figure 3–6: Schematic View of the Screwdriver Model

structure of the model used as an example in this chapter and the next, and later for recognition. A full listing of the model can be found in Appendix B.

Each model part or superpart has its own coordinate frame system in which the positions and orientations of its constituent parts or surfaces are defined. The origin and orientation of a part or superpart coordinate frame is defined, not according to the model coordinate frame, but to that of the superpart of which it forms a part. This allows the construction of each model part to take advantage of the "natural" axes of the object part being modelled, so that axes of symmetry and stretch can represented correctly. The disadvantage is that if the position of a surface in one part relative to a surface in another is required, several computationally expensive coordinate frame transformations are required.

The mapping of a point (x_i, y_i, z_i) in a model part coordinate frame to the point (x_j, y_j, z_j) in a different model part coordinate frame is achieved by the transformation:

$$\begin{pmatrix} x_j \\ y_j \\ z_j \\ 1 \end{pmatrix} = \begin{pmatrix} & & & t_x \\ & R_{pp}(r,s,t) & & t_y \\ & & & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} \tag{3.1}$$

where $R_{pp}(r, s, t)$ is the rotation, specified in terms of rotation, slant and tilt [Fisher, 1986], and t_x, t_y and t_z are the components of the translation between the two coordinate frames.

In the models used in this work, the rotations between the coordinate frames of different parts and between parts and superparts were all multiples of $\pi/2$.

In SMS the model surface primitives were defined in terms of a basic shape (planar, cylindrical, conical or toroidal), the principal curvatures (where applicable) and the boundary of the surface. In the modelling system used here, the model surfaces are allowed to deform to fit the corresponding image surfaces so that the concept of a fixed shape, while useful as a starting point, is not retained during the matching process. Surface patches are defined only in terms of their principal curvatures. The boundaries of the surface primitives used here are not defined.

3.3.1 Representation of Scale

The representation and use of a scale parameter is common in object recognition. Roberts [Roberts, 1965] and Ayache [Ayache & Faugeras, 1986] both used a scale factor that represented the scale of *all* the objects as depicted in the image. Murray [Murray, 1987b] used a factor S that could be used to represent the scale of an individual object. In Grimson's approach [Grimson, 1987] to matching objects to parameterised models, a scale factor s was used to represent the size or scale of each *part* of an object. The scale of different parts of the same object could be different, though they had to be "roughly the same" for a consistent match.

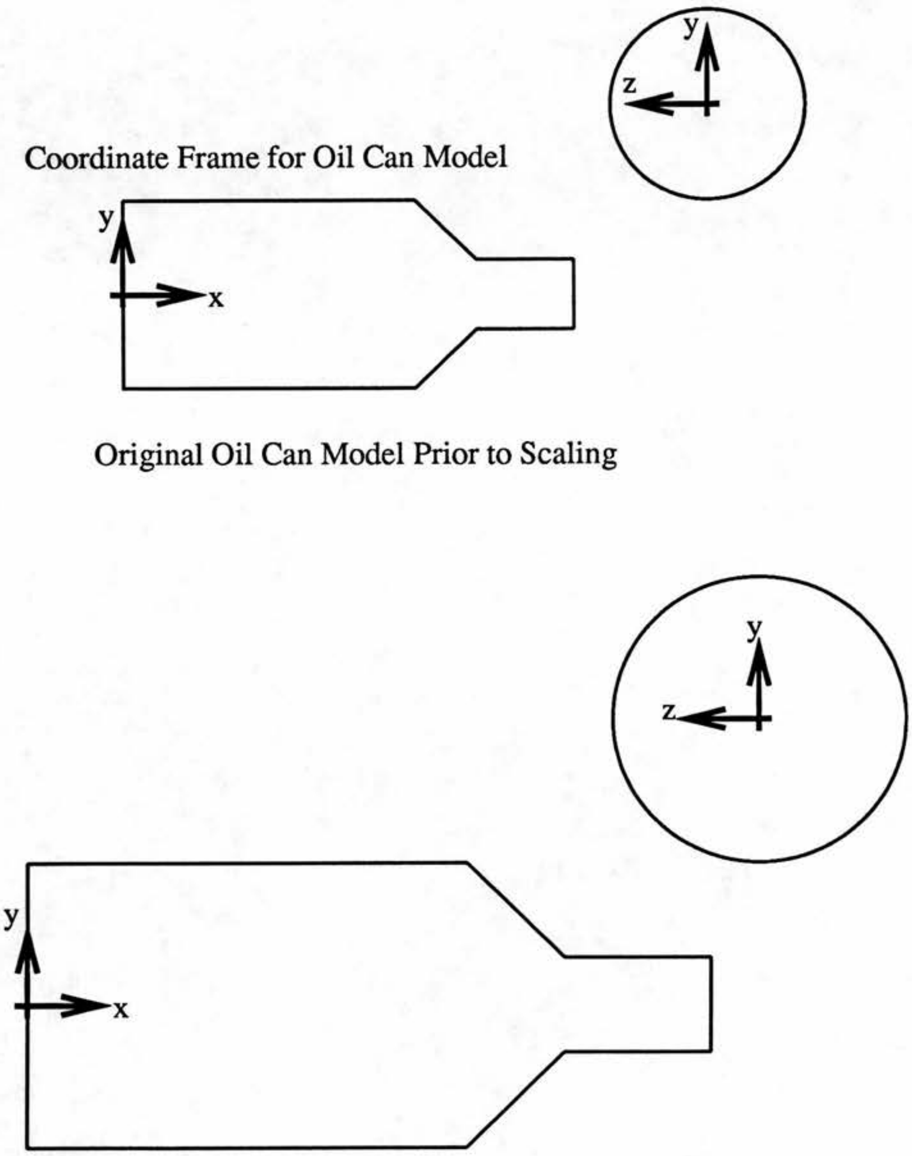
Where the class of objects represented by a model can exhibit changes in overall size or scale, it must be possible to reproduce the effects of this in the model. This enables the degree of fit between an object and corresponding model to be ascertained and also enables the appearance and position in the image of any unmatched model features to be determined. The factor that defines the size of the object relative to the original model is the coefficient of global scale S_g. Because this change in scale is uniform throughout all parts of the object, all parts of the corresponding model must be scaled too.

Where a model part is to be scaled, any point in the original *model* (x_i, y_i, z_i) is transformed to the corresponding point in the scaled model $(x_{g_i}, y_{g_i}, z_{g_i})$ by the standard scaling transformation:

$$\begin{pmatrix} x_{g_i} \\ y_{g_i} \\ z_{g_i} \\ 1 \end{pmatrix} = \begin{pmatrix} S_g & 0 & 0 & 0 \\ 0 & S_g & 0 & 0 \\ 0 & 0 & S_g & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} \tag{3.2}$$

The position of the corresponding point in the image (X_i, Y_i, Z_i) is given by the transformation:

$$\begin{pmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{pmatrix} = \begin{pmatrix} & & & t_{xmw} \\ & R_{mw}(r,s,t) & & t_{ymw} \\ & & & t_{zmw} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{g_i} \\ y_{g_i} \\ z_{g_i} \\ 1 \end{pmatrix} \tag{3.3}$$



Coordinate Frame for Oil Can Model

Original Oil Can Model Prior to Scaling

Oil Can Model Uniformly Scaled in all Dimensions
Global Scale Factor S_g is 1.5

Figure 3–7: The Effect of the Global Scale on a Model

where $R_{mw}(r, s, t)$ and t_{xmw}, $t_{y_{mw}}$ and t_{zmw} are the coordinate frame rotation and translation respectively between the coordinate frame of the model within which the point lies and the world coordinate frame. Figure 3–7 shows the effect of a model subjected to the global scale transform.

This scaling transformation applies equally to the positions of model parts within model superparts and model superparts within the model itself. It also affects the curvatures of any surfaces in the model. Figure 3–8 shows how the global scale of an object is represented in the model.

Representing the global scale in this way such that it is uniform in all directions throughout the model, enables the position and appearance of object features to be predicted from the model without the need for computationally expensive coordinate frame transformations.

The global scale could theoretically take any positive value. Values less than one mean the object is smaller than the model and the model should be "shrunk" accordingly, while values greater than one mean the model should be "enlarged" to fit the object. Generally though, the range of scales in a given class of object can be expected to fall between two limits. Where this is the case, the limits can be specified in the model and used as an extra constraint during the matching process. Because the global scale affects the whole model, its limits are specified only once at the highest level of the model. The example below shows the global scale of the OIL_CAN model being specified as having lower and upper limits of 0.1 and 10.0 respectively. This means that for the class of oil cans represented, the dimensions of an object may vary uniformly by a factor of between 0.1 and 10.0:

```
Name : OIL_CAN                        % Name of the model
Model type : Model                    % Indicates a model
Scale lower limit : 0.1               % Lower limit on scale
Scale upper limit : 10.0              % Upper limit on scale
Stretch lower limits : 0.25 0.25 0.25
Stretch upper limits : 4.00 4.00 4.00
Posn coeffs : 1.0 1.0 1.0
```

```
Model position : 0.0 0.0 0.0
Orient coeffs : 0.8 0.8 0.8
Model orientation : 0.0 0.0 0.0
Number of parts : 2
```

3.3.2 Representation of Stretch

Grimson [Grimson, 1987] used object parts that could "stretch" by a variable amount α along a known axis, termed the "stretch axis", and matched these with a parameterised model. The base case, when $\alpha = 1$, was where there was no stretching of the object part relative to the model. The objects used were imaged as two dimensional objects. Had Grimson used object parts that exhibited both scale and stretch simultaneously this would effectively have given two orthogonal degrees of stretch since the scale factor s was not uniform between parts. However the constraint imposed that the scale of two parts had to be "roughly the same" would have severely constrained the degree of stretching orthogonal to the stretch axis. The extension of this work to the three dimensional objects used here would mean the use of two orthogonal components of stretch and a scale factor. Here, however, the scale factor S_g is defined as constant throughout the model and so cannot be regarded as a third stretch factor for a part. Instead three stretch factors acting along orthogonal axes S_x, S_y and S_z are used in addition to the scale factor.

Where a class of objects can exhibit stretching, it must be possible to reproduce the effects of stretching in the appropriate model part so that the object can be matched against the model parts and the prediction of the appearance and position in the image of any unmatched model features can take place.

Unlike the global scale discussed in the previous section, the amount and direction that a part, superpart or object itself can be stretched may differ between different parts of the same object. Within the local coordinate frame of the p^{th} part of an object, the factors that define the stretch of the object part are the coefficients of stretch S_{xp}, S_{y_p} and S_{zp}. These stretch factors represent *the stretch of the object part or superpart relative to the stretch of the superpart or model that it is a part.* Unlike the global

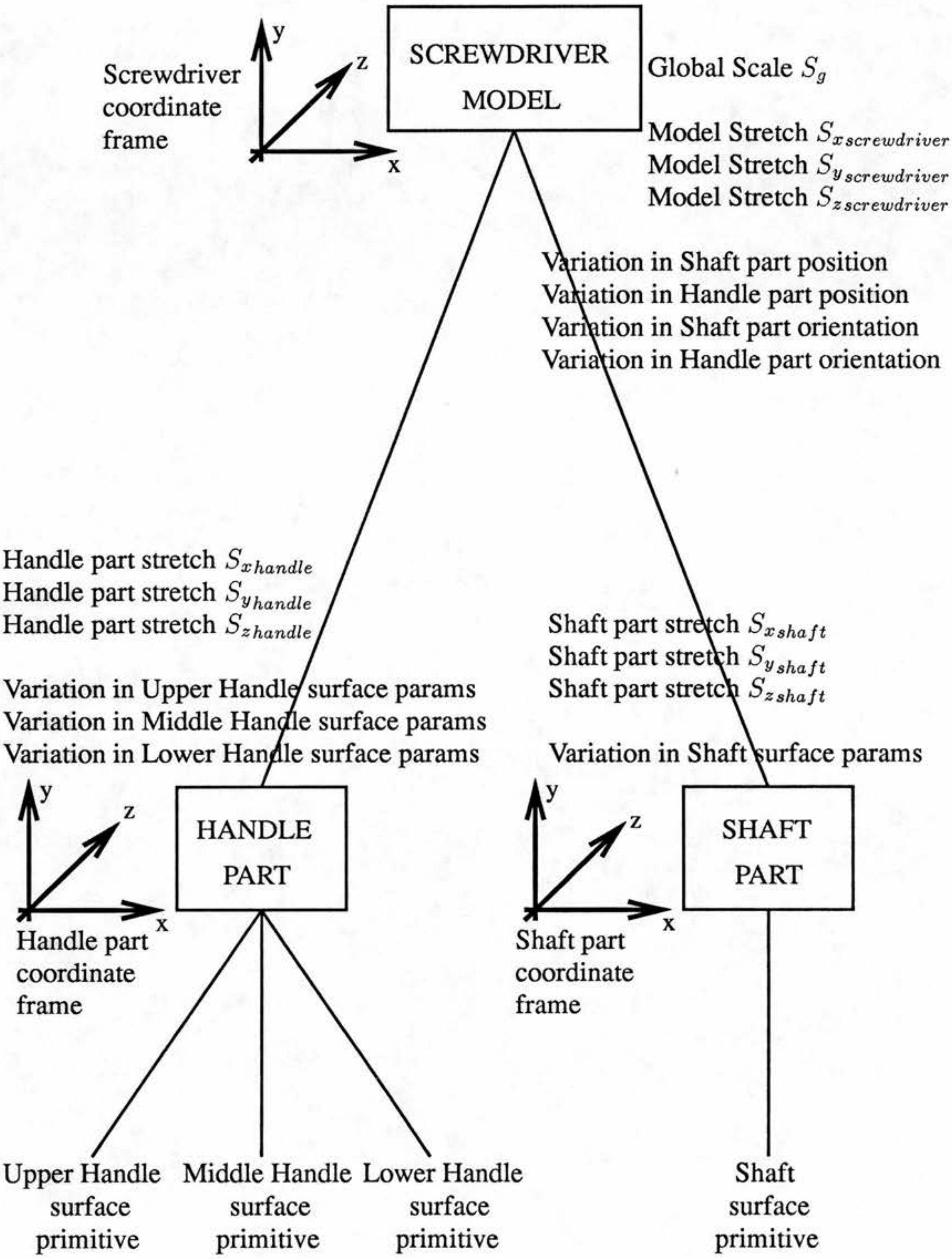
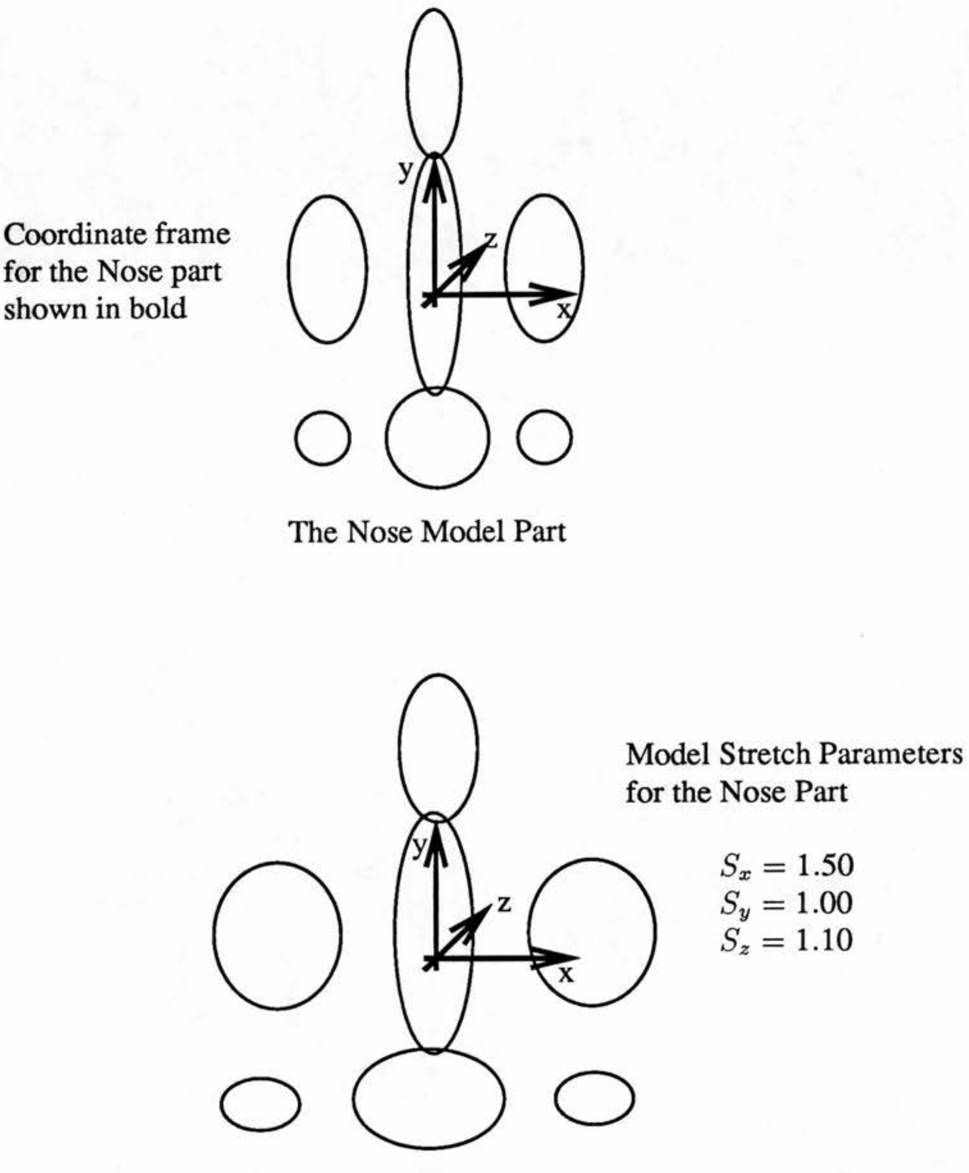

SCREWDRIVER MODEL
Screwdriver coordinate frame
x
y
z
Global Scale S_g
Model Stretch $S_{x_{screwdriver}}$
Model Stretch $S_{y_{screwdriver}}$
Model Stretch $S_{z_{screwdriver}}$
Variation in Shaft part position
Variation in Handle part position
Variation in Shaft part orientation
Variation in Handle part orientation
Handle part stretch $S_{x_{handle}}$
Handle part stretch $S_{y_{handle}}$
Handle part stretch $S_{z_{handle}}$
Variation in Upper Handle surface params
Variation in Middle Handle surface params
Variation in Lower Handle surface params
Shaft part stretch $S_{x_{shaft}}$
Shaft part stretch $S_{y_{shaft}}$
Shaft part stretch $S_{z_{shaft}}$
Variation in Shaft surface params
HANDLE PART
Handle part coordinate frame
SHAFT PART
Shaft part coordinate frame
Upper Handle surface primitive
Middle Handle surface primitive
Lower Handle surface primitive
Shaft surface primitive

Figure 3–8: The Representation of Deformation in the Screwdriver Model



Coordinate frame for the Nose part shown in bold

The Nose Model Part

Model Stretch Parameters for the Nose Part

$S_x = 1.50$
$S_y = 1.00$
$S_z = 1.10$

The Nose Model Part Following the Effects of Stretching

Figure 3–9: The Effect of Stretch on Model Parts

scale, they do *not usually* represent the stretching of the object part relative to the original model part. This point is explained in greater detail in the next chapter. These coefficients of stretch act along the x, y and z axes respectively of the coordinate frame of that model part, so it is important that the axes of the model part coordinate frame be aligned with the "natural" axes of stretch of that object part.

Where a model part is to be stretched, a point in the p^{th} part (x_i, y_i, z_i) is transformed to the corresponding point in the stretched model part (x_{si}, y_{si}, z_{si}) by the transformation:

$$\begin{pmatrix} x_{si} \\ y_{si} \\ z_{si} \\ 1 \end{pmatrix} = \begin{pmatrix} S_{xp} & 0 & 0 & 0 \\ 0 & S_{y_p} & 0 & 0 \\ 0 & 0 & S_{zp} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} \tag{3.4}$$

The transformation given below is only valid for the mapping of a point in a stretched *model coordinate frame* (not a superpart or part coordinate frame) (x_{si}, y_{si}, z_{si}) to the corresponding point in the image (X_i, Y_i, Z_i):

$$\begin{pmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{pmatrix} = \begin{pmatrix} & & & t_{xmw} \\ & R_{mw}(r,s,t) & & t_{y_{mw}} \\ & & & t_{zmw} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{si} \\ y_{si} \\ z_{si} \\ 1 \end{pmatrix} \tag{3.5}$$

Where $R_{mw}(r,s,t)$ and t_{xmw}, $t_{y_{mw}}$ and t_{zmw} are the coordinate frame rotation and translation respectively between the model coordinate frame and the world coordinate frame. Figure 3–9 shows the effects of a model part subjected to a stretching transform.

Model superparts and models themselves may also appear stretched and the positions and orientations of parts within superparts and superparts within models are also obtained by this transformation. Within parts, the curvatures of the component surfaces can be affected. Unlike the global scale S_g, the coefficients of stretch are specific only to a model part so the exact values used in the transformation will vary between parts. Figure 3–8 shows how model and part stretch are represented in the model. It is important to note that the coefficients of stretch for a superpart or model are specific only to that superpart or model and are not necessarily dependent on the stretch coefficients of their constituent parts or superparts. This aspect is also dealt with in the next chapter.

The equivalent translation for a point in a stretched *superpart* or *part* would need to take into account the stretch of any superparts higher in the model hierarchy and the stretch and scale of the model itself. The combining of the effects of global scale and all coefficients of stretch from higher levels in the hierarchy are represented by the parameters of combined scale and stretch S_{gx}, S_{gy} and S_{gz}. Figure 3–10 shows how this is applied to the screwdriver model. Because the coordinate frames of the two model parts are aligned with that of the model, the combined scale and stretch of a part, for example the handle, is simply the product of the global scale of the model, the stretch of the model and the stretch of the handle itself. The representation of stretch in this manner and its use during the recognition process is discussed in the next chapter.

The coefficients of stretch, like the global scale, can take any positive value but can generally be expected to fall within a given range for a particular class of object. As before, acceptable limits for the values of stretch can be stored in the model and used as constraints in the matching process. Because the stretch factors for a given part are consistent only for that part then each part, superpart and model must have its own limits. During recognition, the parameters that are used are those of the combined effects of scale and stretch and it is these limits that are stored in the model. The example below shows the coefficients of combined scale and stretch for the NOSE model part being specified as all having lower and upper limits of 0.5 and 2.0 respectively. This means that the corresponding object part may be *scaled and stretched* relative to the model part along each of the model coordinate frame axes by (possibly different) factors of between 0.5 and 2.0:

```
Name : NOSE                                % Name of part
Part type : Part                           % Identified as a part
Stretch lower limits : 0.5 0.5 0.5         % Lower limits of three
                                           % components of stretch
Stretch upper limits : 2.00 2.00 2.00      % Upper limits of three
                                           % components of stretch
```

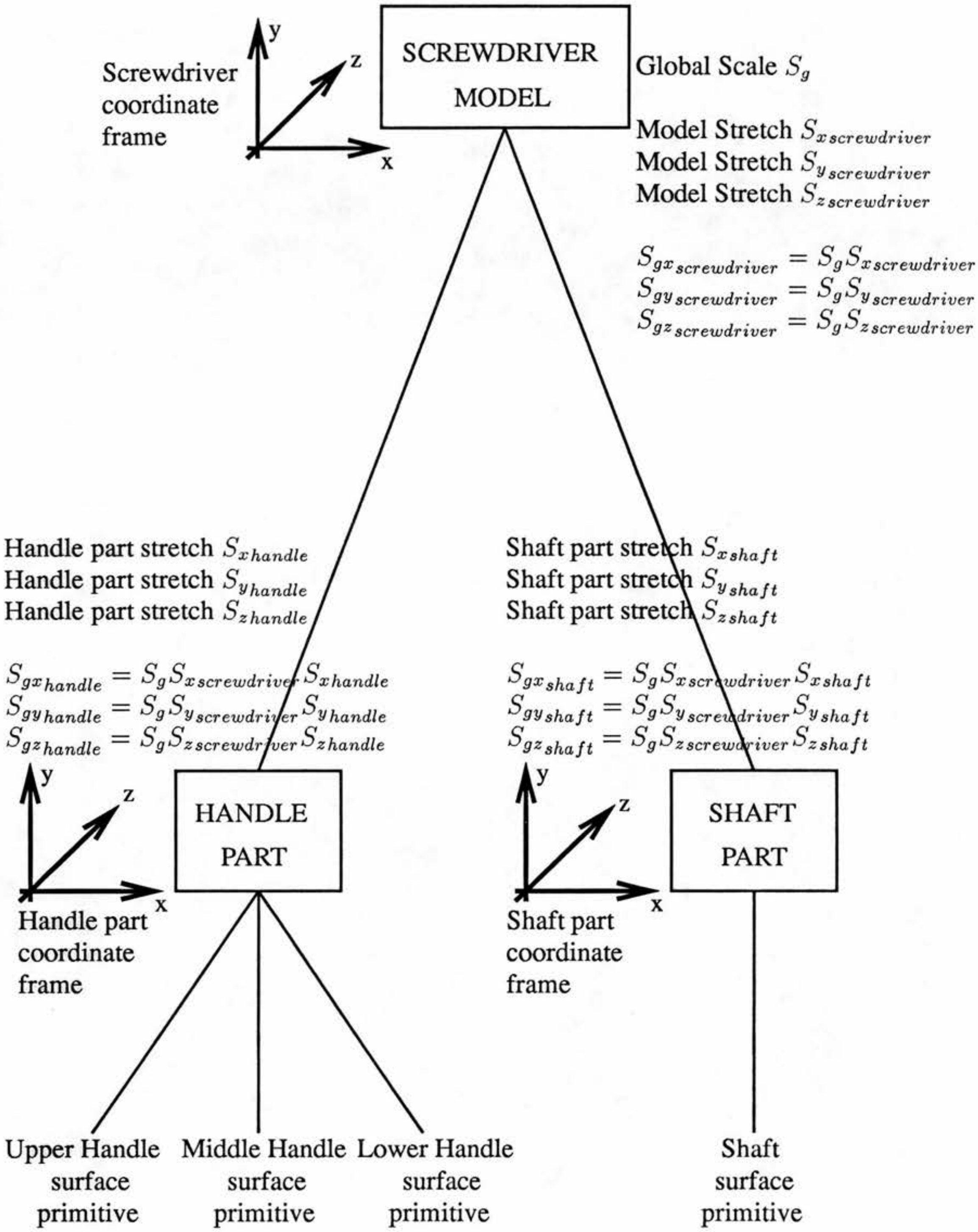

Screwdriver coordinate frame
y
z
x
SCREWDRIVER MODEL
Global Scale S_g
Model Stretch $S_{x\,screwdriver}$
Model Stretch $S_{y\,screwdriver}$
Model Stretch $S_{z\,screwdriver}$
$S_{gx\,screwdriver} = S_g S_{x\,screwdriver}$
$S_{gy\,screwdriver} = S_g S_{y\,screwdriver}$
$S_{gz\,screwdriver} = S_g S_{z\,screwdriver}$
Handle part stretch $S_{x\,handle}$
Handle part stretch $S_{y\,handle}$
Handle part stretch $S_{z\,handle}$
$S_{gx\,handle} = S_g S_{x\,screwdriver} S_{x\,handle}$
$S_{gy\,handle} = S_g S_{y\,screwdriver} S_{y\,handle}$
$S_{gz\,handle} = S_g S_{z\,screwdriver} S_{z\,handle}$
Shaft part stretch $S_{x\,shaft}$
Shaft part stretch $S_{y\,shaft}$
Shaft part stretch $S_{z\,shaft}$
$S_{gx\,shaft} = S_g S_{x\,screwdriver} S_{x\,shaft}$
$S_{gy\,shaft} = S_g S_{y\,screwdriver} S_{y\,shaft}$
$S_{gz\,shaft} = S_g S_{z\,screwdriver} S_{z\,shaft}$
y
z
x
HANDLE PART
Handle part coordinate frame
y
z
x
SHAFT PART
Shaft part coordinate frame
Upper Handle surface primitive
Middle Handle surface primitive
Lower Handle surface primitive
Shaft surface primitive

Figure 3–10: The Representation of the Combined Effects of Deformation in the Screwdriver Model

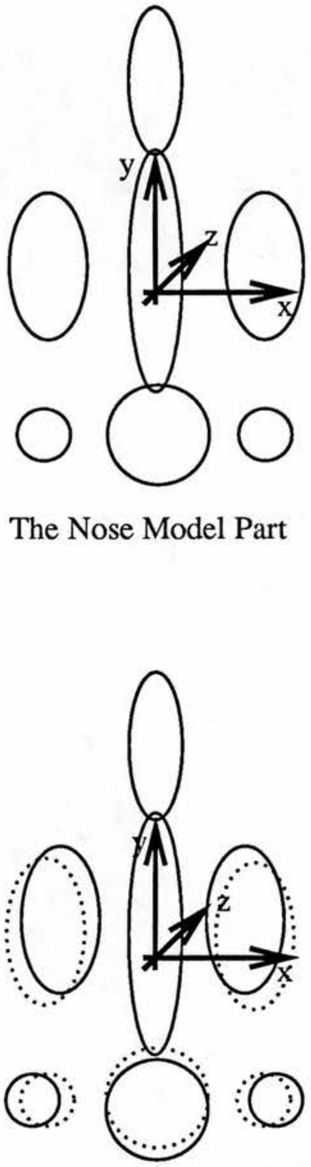
```
Posn coeffs : 1.0 1.0 1.0              %)
Part position : 0.0 -8.0 -46.0         %) These explained in
Orient coeffs : 0.8 0.8 0.8            %) section 3.3.3
Part orientation : 0.0 0.0 0.0         %)
Number of parts : 5
```

The example below shows the limits of stretch for the HUMAN_HEAD model. Although these stretch value limits are defined at the model level they can affect the positions of the model's two surfaces – the top of the head and the back of the head and its superpart – the face.

```
Name : HUMAN_HEAD                           % Model name
Model type : Model                          % Identified as a model
Scale lower limit : 0.5                     % Lower limit of scale
Scale upper limit : 2.0                     % Upper limit of scale
Stretch lower limits : 0.75 0.75 0.75 % Lower limits of three
                                            % components of stretch
Stretch upper limits : 1.50 1.50 1.50 % Upper limits of three
                                            % components of stretch
Posn coeffs : 1.0 1.0 1.0
Model position : 0.0 0.0 0.0
Orient coeffs : 0.8 0.8 0.8
Model orientation : 0.0 0.0 0.0
Number of parts : 3
```

3.3.3 Representation of Individual Variation

Previous work that has dealt with objects that show scaling or stretching has assumed that there is an exact transformation for each feature within the object or object part. The position of an object feature can therefore be predicted exactly, though usually measurement errors mean that its position must be predicted to lie within an area or a volume. The use here of small variations or perturbations is somewhat similar to the



Coordinate frame for the Nose part shown in bold

The Nose Model Part

Original model positions shown dotted

Small variations in position are unique to each surface

The Nose Model Part Following the Effects of Individual Variation

Figure 3–11: The Effect of Variation on Surfaces in a Model Part

"rubber mask" technique [Fischler & Elschlager, 1973]. The use of different "spring rates" in that work is similar to the weighting coefficients used here.

The models that represent classes of objects that can exhibit individual variation should also be able to reproduce the effects of individual variation. Unlike the previous examples of changes in scale and stretch, the estimate of individual variation of a superpart, part or surface is not always required before matching can take place between an example of the object class and the model. The effects of individual variation are used in prediction and matching only where an object contains one or more identical features. In this case the variation present in one is assumed to be present in the other. Variation is however useful in determining the "degree of fit" between a feature in the deformed model and an object feature in the image.

Where some feature of the model shows individual variation, in this case the position of the j^{th} model surface (x_j, y_j, z_j), then the position in the deformed model (x_{vj}, y_{vj}, z_{vj}) is given by the transformation:

$$\begin{pmatrix} x_{vj} \\ y_{vj} \\ z_{vj} \\ 1 \end{pmatrix} = \begin{pmatrix} V_{xj} & 0 & 0 & 0 \\ 0 & V_{yj} & 0 & 0 \\ 0 & 0 & V_{zj} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_j \\ y_j \\ z_j \\ 1 \end{pmatrix} \tag{3.6}$$

In this instance a simple translation could just as easily have been used to represent variation in position but a multiplicative factor was used for consistency with the previous deformations and to simplify the representation of the scale and stretch recovery problem in the next chapter.

The position of the corresponding point in the image (X_j, Y_j, Z_j) is given by the transformation:

$$\begin{pmatrix} X_j \\ Y_j \\ Z_j \\ 1 \end{pmatrix} = \begin{pmatrix} & & & t_{xpw} \\ & R(r,s,t) & & t_{ypw} \\ & & & t_{zpw} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{vj} \\ y_{vj} \\ z_{vj} \\ 1 \end{pmatrix} \tag{3.7}$$

where $R_{pw}(r, s, t)$ and t_{xpw}, $t_{y_{pw}}$ and t_{zpw} are the coordinate frame rotation and translation respectively between the model part coordinate that contains the surface and the world coordinate frame. The variation parameters V_{xj}, V_{y_j} and V_{zj} are unique to the j^{th} surface. Figure 3–11 shows the effect of variations on features of a part.

The variation shown here as an example is a variation in position. Variations may occur in any of the model properties and unlike the scale or stretch deformations the variation shown by one property is *not* expected to be shown by another. Properties of a surface, part, superpart or model may all show variation but any variation affects only that property. It does not affect the properties of the parts above or below it in the model hierarchy. Figure 3–8 shows the way in which variation is represented in the model.

Usually a class of objects will exhibit variation in a large number, if not all, of its features. To predict the position and appearance of each model feature in this way would be computationally expensive but fortunately this is only necessary where a model contains identical surfaces or parts.

Previously, the acceptable values for the global scale and stretch were specified by definite limits in the model. Individual variations are by definition very small (variations in positions are generally less than ten percent) and so the idea of a finite limit is of limited use. Instead, rather than use a fixed limit for the variation, a "weighting coefficient" is used. This allows the individual variation of some property of the object to take any positive value but as the variation grows larger, the penalty imposed by the weighting coefficient also becomes larger. The use of these weighting coefficients during the recognition process is described in later chapters.

The example below shows the weighting coefficients associated with a surface primitive:

```
curv_1 : 0.006667     % First princ. curv. of surface
curv_2 : 0.006667     % Second curv. of surface
curv_coeff : 1.0      % Weighting coeff of any
                      % variation in curvature
```

The curv_coeff is the weighting coefficient for variations in the two curvature values. These weighting values apply not to the difference between the stated curvatures of the model surface and the corresponding image surface, but to the difference between the curvatures of the model surface after scaling and stretching and the object surface.

The example below shows the weighting coefficients associated with the position and orientation of the surface:

```
Name : Back_of_Skull                % Name of part
Part type : Primitive               % Identified as a surface
Posn coeffs : 1.0 1.0 1.0           % Weighting coefficients for
                                    % variations in translation.
Part position : 0.0 -8.0 -46.0      % Translation of surface
                                    % in part coordinate frame
Orient coeffs : 0.8 0.8 0.8         % Weighting coefficients for
                                    % Variation in rotation.
Part orientation : 0.0 0.0 0.0      % Rotation of surface
                                    % in part coordinate frame.
Primitive id : 0                    % Identifies which surface
                                    % is the primitive.
```

Since this specifies only a single model surface, the global scale and part stretch limits are not specified.

3.4 Conclusion

This chapter describes the construction of a deformable three dimensional parameterised model for use in computer recognition. Three sets of parameters are defined, the global scale that defines the overall size of the model, the three orthogonal components of stretch that define the elongation or compression of a model part and the individual variation that allows for the small perturbations in the model not accounted for by the scale or stretch. These parameters reflect the hierarchical structure of the model. The global scale is defined at the highest level of the model. The stretch parameters of each part are defined not according to the model, but in terms of those of the superpart above in the same way as the coordinate frames are defined. At the lowest level, the individual variations for each component of a part are represented.

In the next chapter it is shown how the deformation parameters are combined and how they may be resolved as part of the recognition process.

Chapter 4

Resolving the Deformation Parameters in the Image

4.1 Introduction

When constructing parameterised models for object recognition, the parameters used must not only be capable of reproducing in the model the effects observed in the object, but must also be easily and stably recoverable from the image during the recognition process.

The deformation classes of global scale, part stretch and individual variation as defined in the previous chapter often occur together. Any deformation of an object or object part is usually due to the combined effects of all three types. Furthermore, the degree and axes of stretch in a model or superpart will have an influence on the degrees of stretch in their constituent superparts or parts. In this chapter the interrelationship of each type is defined and the approach by which the effects of each can be resolved is outlined.

4.2 The Interrelationship of the Deformation Parameters

In the previous chapter, the deformation parameters of global scale, part stretch and individual variation were introduced. Although each is represented by independent parameters their effects are interrelated.

Starting with an object o as a whole, it will have a global scale S_g, stretch parameters S_{xo}, S_{y_o} and S_{z_o} and individual parameters of variation in the properties of the parts and superparts that make up the object. All three types are normally present in the object and the deformation exhibited by the object is the combined effect of each. Before matching between model and object can take place, the corresponding model must be deformed in the same way as the object. A point in the coordinate frame of the original model (x_i, y_i, z_i) will map to the corresponding point in the deformed *model* (x_{oi}, y_{oi}, z_{oi}) by the transformation:

$$\begin{pmatrix} x_{oi} \\ y_{oi} \\ z_{oi} \\ 1 \end{pmatrix} = \begin{pmatrix} S_g S_{xo} V_{xi} & 0 & 0 & 0 \\ 0 & S_g S_{y_o} V_{y_i} & 0 & 0 \\ 0 & 0 & S_g S_{zo} V_{zi} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} \tag{4.1}$$

where V_{xi}, V_{y_i} and V_{zi} are the individual variations in the position of the point.

At the next level of detail in the object, an object superpart sp may appear scaled and stretched relative to the corresponding *model superpart*. There is also the individual variation of its constituent superparts, parts and surfaces. The scaling of the superpart is the same as that of the object, the global scale factor S_g. The stretch of the superpart is defined by its independent stretch parameters S_{xsp}, $S_{y_{sp}}$ and S_{zsp} that represent *not* the stretch of the object superpart sp relative to the original model superpart, but *the stretch of the object superpart relative to the next level up in the hierarchy*, in this case the object o.

The same philosophy is applied to the stretch of any component superparts or parts. The stretch parameters of an object superpart or part reflect its stretch relative to the stretch of the object or superpart of which it forms part rather than its stretch relative to the corresponding model superpart or part.

For example, the screwdriver is modelled as shown in Figure (3–6). The corresponding object is therefore expected to be an object consisting of two parts, one of three surfaces and one with one surface. At the highest level in the representation is the global scale S_g and the stretch components of the object $S_{x\,screwdriver}$, $S_{y\,screwdriver}$ and $S_{z\,screwdriver}$. If the screwdriver was stretched along its length (represented by the model x–axis) by a factor of two, then $S_{x\,screwdriver}$ would become two, while $S_{y\,screwdriver}$ and $S_{z\,screwdriver}$ would remain unchanged.

Where the screwdriver as a whole appears stretched then it is expected that this stretch will also be exhibited by its two parts. If this is the case then the stretch parameters of each part $S_{x\,handle}$, $S_{y\,handle}$ and $S_{z\,handle}$ and $S_{x\,shaft}$, $S_{y\,shaft}$ and $S_{z\,shaft}$ will be *unity* since they represent *the stretching of the part relative to the level above in the hierarchy*, in this case the object. To determine the stretch of each object part *relative to the corresponding model part* would therefore require the *combining* of the stretch of the part, the stretch of the screwdriver and any global scale term. For the screwdriver this is a simple process. The coordinate frames of both parts are the same as that of the model so the parameters of combined scale and stretch $S_{gx\,handle}$, $S_{gy\,handle}$ and $S_{gz\,handle}$ and $S_{gx\,shaft}$, $S_{gy\,shaft}$ and $S_{gz\,shaft}$ are simply the products of global scale, object stretch and part stretch (see Figure (3–10)).

If the screwdriver was stretched along its axis by a factor of two but this stretching was not shared by one of its parts, for example the shaft, then the component of stretch $S_{x\,shaft}$ would be 0.5. The combined stretch component would therefore be unity, the object part showing *no stretch relative to the model* along that axis.

Returning to the more general case of an object o and a superpart sp, the stretch of the object superpart relative to the model superpart is a function of the global scale S_g, the stretch of the object o S_{xo}, S_{y_o} and S_{zo} and the stretch of the superpart sp $S_{x\,sp}$, $S_{y_{sp}}$ and $S_{z\,sp}$. A point in the original model superpart (x_i, y_i, z_i) is mapped to the point in the deformed superpart $(x_{sp_i}, y_{sp_i}, z_{sp_i})$ by the transformation:

$$\begin{pmatrix} x_{sp_i} \\ y_{sp_i} \\ z_{sp_i} \\ 1 \end{pmatrix} = \begin{pmatrix} S_{x\,sp}V_{xi} & 0 & 0 & 0 \\ 0 & S_{y_{sp}}V_{y_i} & 0 & 0 \\ 0 & 0 & S_{z\,sp}V_{zi} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} \tag{4.2}$$

where V_{xi}, V_{y_i} and V_{zi} are the individual variations in the position of the point.

If the superpart coordinate frame to model coordinate frame transformation is a rotation $R_{spm}(r,s,t)$ and translation $t_{x\,spm}$, $t_{y_{spm}}$ and $t_{z\,spm}$ then the point in the deformed superpart $(x_{sp_i}, y_{sp_i}, z_{sp_i})$ is mapped to the corresponding point in the deformed model (x_{oi}, y_{oi}, z_{oi}) by the transformation:

$$\begin{pmatrix} x_{oi} \\ y_{oi} \\ z_{oi} \\ 1 \end{pmatrix} = \begin{pmatrix} S_gS_{xo}V_{x\,sp} & 0 & 0 & 0 \\ 0 & S_gS_{y_o}V_{y\,sp} & 0 & 0 \\ 0 & 0 & S_gS_{z_o}V_{z\,sp} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} & & & t_{x\,spm} \\ & R_{spm}(r,s,t) & & t_{y_{spm}} \\ & & & t_{z\,spm} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{sp_i} \\ y_{sp_i} \\ z_{sp_i} \\ 1 \end{pmatrix} \tag{4.3}$$

where $V_{x\,sp}$, $V_{y_{sp}}$ and $V_{z\,sp}$ are the individual variations in the position of the superpart sp within the object o coordinate frame.

The representation of part stretch in this manner reflects the hierarchical structure of the model with parts defined in terms of the superparts of which they form part, rather than their relationship to the model as a whole.

4.3 Resolving the Effects of Each Set of Deformation Parameters

Although the effects of the global scale, each level of part stretch and individual variation are closely interrelated, each must be accurately resolved if accurate matching and feature prediction is to take place in the recognition process.

In resolving the effects of scale and stretch, two basic, but very important, criterion are used:

1. The deformation of an object should be explained as far as possible by its global scale.

2. Within a part, any residual deformation should be explained as far as possible by part stretch.

The stretch parameters of an object part reflect its stretch relative to the stretch of its object superpart rather than its stretch relative to the corresponding model part. This representation is therefore of limited use when seeking object to model correspondences or in the prediction of features in the image where the stretch between object part and model part is required. As defined earlier the stretch of an object part, here denoted p, relative to the corresponding model part is obtained by combining the effects of global scale and the stretch in *all* levels of hierarchy down to the part. The *combined scale and stretch* parameters of the part p, S_{gx_p}, S_{gy_p} and S_{gz_p} are used to represent this.

A point (x_i, y_i, z_i) in the model part is mapped to the the corresponding point in the deformed model part (x_{ci}, y_{ci}, z_{ci}) by the transformation:

$$\begin{pmatrix} x_{ci} \\ y_{ci} \\ z_{ci} \\ 1 \end{pmatrix} = \begin{pmatrix} S_{gx_p}V_{xi} & 0 & 0 & 0 \\ 0 & S_{gy_p}V_{y_i} & 0 & 0 \\ 0 & 0 & S_{gz_p}V_{zi} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} \tag{4.4}$$

where V_{xi}, V_{y_i} and V_{zi} are the individual variations in position.

The relationship between the point in the deformed model part (x_{ci}, y_{ci}, z_{ci}) and the corresponding point in the object part (X_i, Y_i, Z_i) is now simply:

$$\begin{pmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{pmatrix} = \begin{pmatrix} & & & t_{xpw} \\ & R_{pw}(r,s,t) & & t_{y_{pw}} \\ & & & t_{zpw} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{ci} \\ y_{ci} \\ z_{ci} \\ 1 \end{pmatrix} \tag{4.5}$$

where $R_{pw}(r, s, t)$ is the rotation and t_{xpw}, $t_{y_{pw}}$ and t_{zpw} is the translation between the model part coordinate frame and the world coordinate frame.

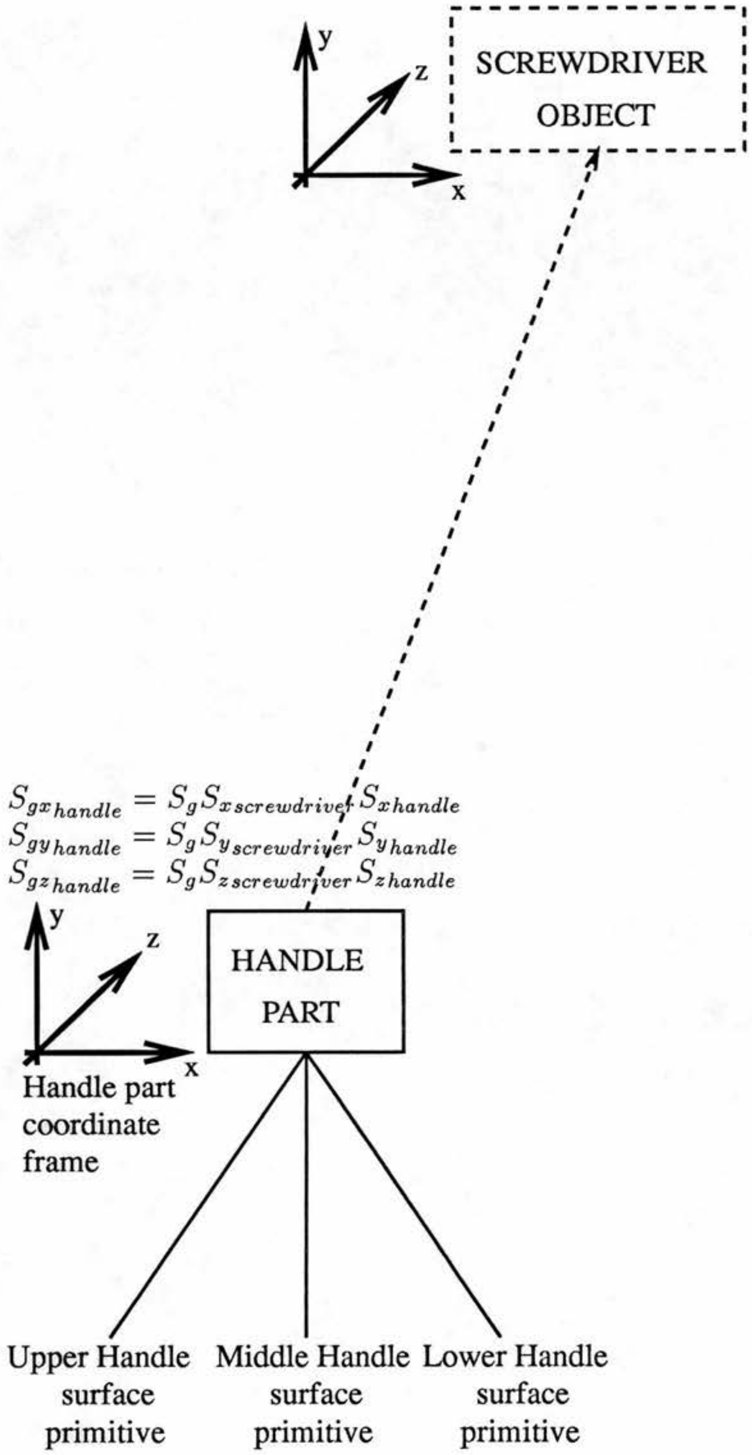

y
z
x
SCREWDRIVER OBJECT
$S_{gx_{handle}} = S_g S_{x_{screwdriver}} S_{x_{handle}}$
$S_{gy_{handle}} = S_g S_{y_{screwdriver}} S_{y_{handle}}$
$S_{gz_{handle}} = S_g S_{z_{screwdriver}} S_{z_{handle}}$
y
z
x
HANDLE PART
Handle part coordinate frame
Upper Handle surface primitive
Middle Handle surface primitive
Lower Handle surface primitive

Figure 4–1: The Combined Scale and Stretch of the Screwdriver Shaft Part

Within an object part, the positions and appearances of the constituent surfaces are governed by the combined effects of scale, all levels of part stretch and the individual variation of the surfaces themselves as defined by the transformation given in Equation (4.4). Obviously the effect of a combined scale and stretch can be the same as that of consistent individual variations in the positions and appearances of the surfaces. However, it is assumed that individual variations are small and any displacements in the positions and appearances of the surfaces should be explained as far as possible by a combined scale and stretch operation. Since all surfaces in the part are affected by the same scale and stretch, the best values for S_{gx_p}, S_{gy_p} and S_{gz_p} are those that minimise the individual variations for that part. Figure (4–1) shows the combined scale and stretch parameters for the screwdriver handle part $S_{gx_{handle}}$, $S_{gy_{handle}}$ and $S_{gz_{handle}}$ as calculated from the three component surfaces.

At the next level up in the hierarchy, an object superpart sp will have its own combined parameters of scale and stretch $S_{gx_{sp}}$, $S_{gy_{sp}}$ and $S_{gz_{sp}}$. As before the best values for these are those that minimise the individual variations in the constituent parts and surfaces. The combined parameters include the effects of global scale as well as stretch. The effect of a scaling operation is the same as a uniform stretching along all axes. The important assumption made here is that the effects of stretch are small relative to the effects of scale and any displacements in position and appearance of the component parts and surfaces should be explained as far as possible by the global scale factor S_g. Since all components of the superpart are affected by the global scale, the best value is that which minimises the effects of part stretch such that any residual stretch is close to unity. With the global scale estimated and the residual stretch and individual variations of the superpart known, the combined scale and stretch factors for its component parts S_{gx_p}, S_{gy_p} and S_{gz_p} can be broken down into global scale and the stretch parameters of that part. Figure (4–2) shows the situation following recognition of the screwdriver object and estimation of the global scale. The combined scale and stretch parameters of the object $S_{gx_{screwdriver}}$, $S_{gy_{screwdriver}}$ and $S_{gz_{screwdriver}}$ can be resolved by simply dividing through by the global scale S_g. Since the coordinate frame of the object is the same as that of the shaft part, the stretch of the part can be obtained by dividing by the global scale and the stretch of the object.

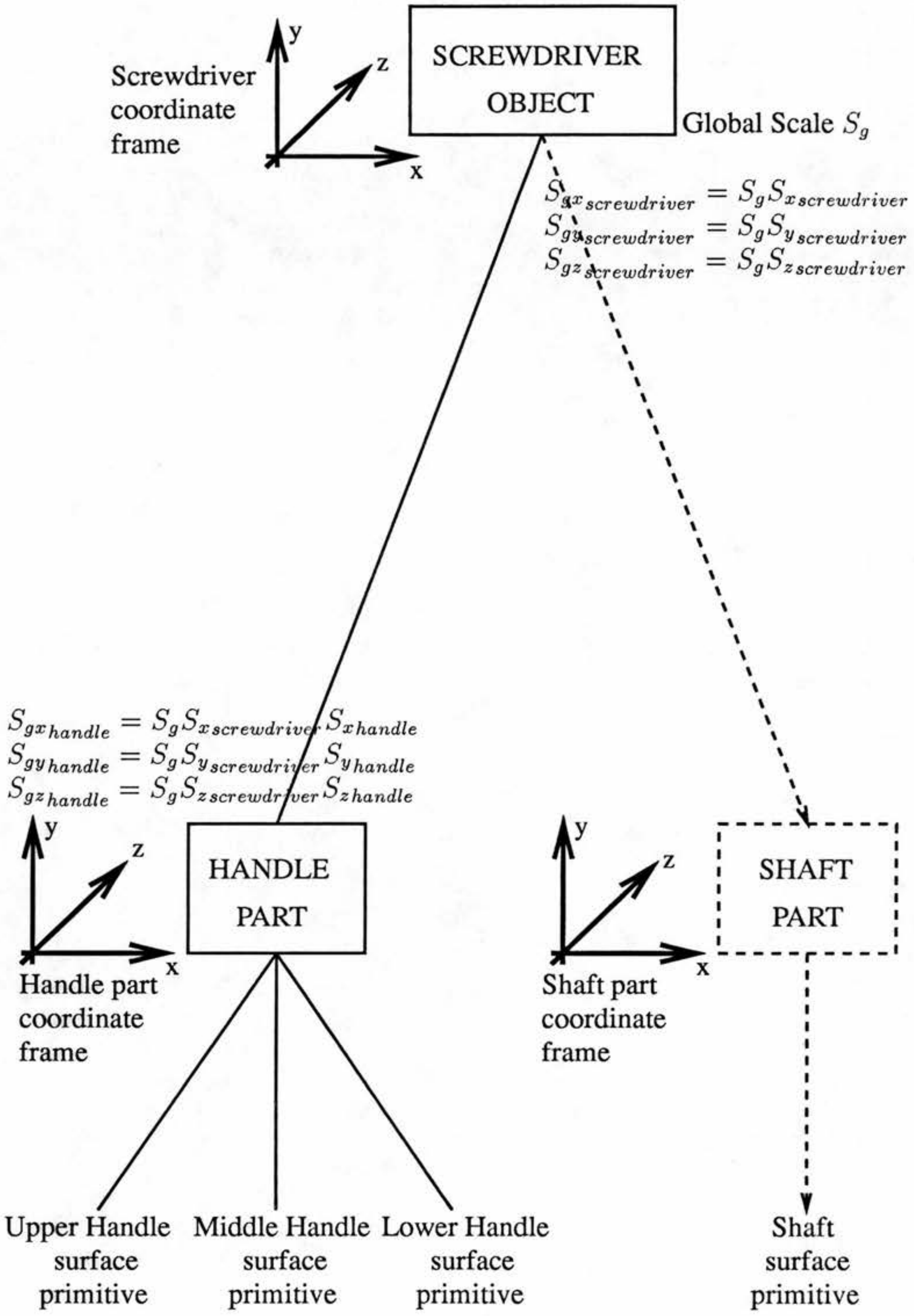

Screwdriver coordinate frame
y
z
x
SCREWDRIVER OBJECT
Global Scale S_g
$S_{gx\,screwdriver} = S_g S_{x\,screwdriver}$
$S_{gy\,screwdriver} = S_g S_{y\,screwdriver}$
$S_{gz\,screwdriver} = S_g S_{z\,screwdriver}$
$S_{gx\,handle} = S_g S_{x\,screwdriver} S_{x\,handle}$
$S_{gy\,handle} = S_g S_{y\,screwdriver} S_{y\,handle}$
$S_{gz\,handle} = S_g S_{z\,screwdriver} S_{z\,handle}$
y
z
x
HANDLE PART
Handle part coordinate frame
y
z
x
SHAFT PART
Shaft part coordinate frame
Upper Handle surface primitive
Middle Handle surface primitive
Lower Handle surface primitive
Shaft surface primitive

Figure 4–2: The Combined Scale and Stretch of the Screwdriver Object

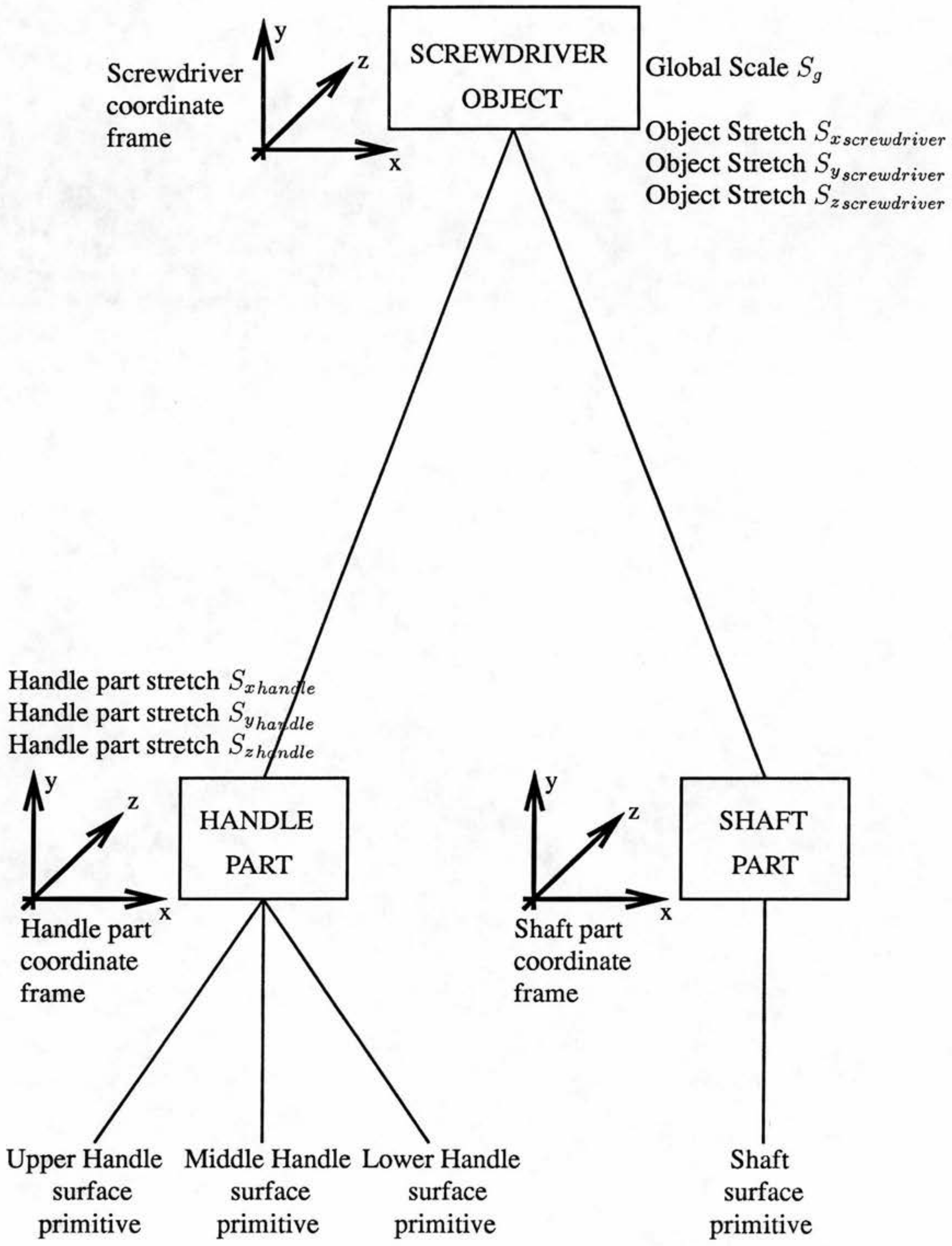

Screwdriver coordinate frame
y
z
x
SCREWDRIVER OBJECT
Global Scale S_g
Object Stretch $S_{x\,screwdriver}$
Object Stretch $S_{y\,screwdriver}$
Object Stretch $S_{z\,screwdriver}$
Handle part stretch $S_{x\,handle}$
Handle part stretch $S_{y\,handle}$
Handle part stretch $S_{z\,handle}$
HANDLE PART
Handle part coordinate frame
SHAFT PART
Shaft part coordinate frame
Upper Handle surface primitive
Middle Handle surface primitive
Lower Handle surface primitive
Shaft surface primitive

Figure 4–3: Final Resolution of the Scale and Stretch of the Screwdriver

This process is repeated up through the hierarchy. At each level, the estimate of the global scale is updated and the stretch components and individual variations at that level determined. These are passed back down through the hierarchy, enabling the components of the previously combined scale and stretch factors to be broken down. The process finishes when at the top of the hierarchy of a fully recognised object, the final estimate of the global scale and the stretch of the object and its parts is made (see Figure (4–3)).

4.4 Examples

The "large" screwdriver shown in Figure (A–9) and the "small" screwdriver shown in Figure (A–8) exhibit both scaling and stretching relative to the "standard" screwdriver shown in Figure (A–7).

The dimensions of the screwdrivers were measured by hand and the parameters of scale and stretch calculated. In the cases below, any stretching along the length of the screwdriver is represented by S_x and across its width by S_z.

For the small screwdriver the combined scale and stretch parameters for the handle part $S_{gx_{handle}}$ and $S_{gz_{handle}}$ are calculated to be 0.538 and 1.550 respectively. The geometric average can be calculated to give a first estimate of the global scale as 0.913. At the next level in the model hierarchy, the combined scale and stretch values for the whole screwdriver $S_{gx_{screwdriver}}$ and $S_{gz_{screwdriver}}$ are estimated to be 0.566 and 1.230 respectively. The updated estimate of the global scale becomes 0.873. Removing the global scale component from the combined scale and stretch factors for the screwdriver resolves the stretch parameters of the screwdriver $S_{x_{screwdriver}}$ and $S_{z_{screwdriver}}$ to be 0.649 and 1.409 respectively. Removing both the global scale and the stretch of the screwdriver from the combined scale and stretch parameters of the handle part gives the values of the part stretch parameters $S_{x_{handle}}$ and $S_{z_{handle}}$ as 0.950 and 1.26 respectively.

These values show that the small screwdriver is (not surprisingly) smaller than the "standard" screwdriver. It shows distinct contraction along its length but enlargement across its width. The stretching of the handle part is similar to the stretching of the whole. The handle shows almost no stretching along its length and only slight stretching along its width relative to the stretched object.

For the large screwdriver the global scale is estimated to be 1.264 indicating the screwdriver to be larger than standard. The stretch of the screwdriver itself is small, $S_{x\,screwdriver}$ is 1.194 and $S_{z\,screwdriver}$ is 0.981 indicating a slight stretching along its length. The stretch of the handle part is also relatively small, $S_{x\,handle}$ is 0.762 and $S_{z\,handle}$ is 0.958. This shows that the elongation of the screwdriver along its axis is shown by the shaft part but not by the handle part.

These predicted values are compared with the values estimated during the recognition process in the model selection and model verification chapters.

4.5 Conclusion

In this chapter the interaction of the three types of deformation – the global scale, the part stretch and the individual variation have been defined. The global scale affects all components of a deformed object equally. The stretch of a part affects the stretch values of all component parts in a hierarchical manner. The individual variation affects only that individual feature to which it belongs.

This chapter also proposes the approach by which the effects of each type of deformation can be resolved as part of the recognition process. The two basic assumptions are that as far as possible the deformation of an object should be explained by the global scale and that, within a part, any residual deformation should be explained as far as possible by part stretch.

Chapter 5

Object Description

5.1 Introduction

The input to this recognition system is in the form of three dimensional, or range, data of an object's surface. In the work described here, a laser range finder is used for convenience, but any method that can produce dense sampling of the distance to an object's surface – such as a stereo range finder, an optical flow system or shape from shading – would be suitable.

The raw range data as obtained from the range finder is not immediately suitable for use with the recognition process. It must first be smoothed and then segmented into surface patches, or regions, consistent with the significant features of the object and the features represented in the corresponding model. Once segmented, the parameters of each surface patch, such as its position and area, must be determined.

The segmentation of the raw range data is performed by grouping adjacent points of uniform curvature. Once the boundaries of these regions have been established the parameters of each region are found using nearest neighbour operations as an aid to both speed and efficiency.

The representation used to describe object surfaces is identical to that used to describe the model surfaces. This ensures that during the matching process the comparison between model and object surfaces is straightforward.

5.2 The Object Range Data

The three dimensional data that forms the input to this recognition process is obtained by the use of a laser striper. The images used here are either obtained using the Edinburgh striper system developed by the vision group at the Department of Artificial Intelligence [Naidu *et al*, 1990] and [Naidu & Fisher, 1990] or from the National Research Council of Canada system [Rioux & Cournoyer, 1988]. An object is placed on a table that can be accurately positioned along a single axis (by convention the i–axis of the image coordinate system) by a calibrated stepper motor. A laser beam is focused using a cylindrical lens to form a stripe of laser light across the table orthogonal to the axis of motion (by convention the j–axis). To obtain the range data the object is moved under the laser stripe in a series of small steps and the vertical range to the object's surface sampled along the stripe at each step. In the NRCC system, it is the camera that is moved rather than the object and the laser stripe is formed by a rotating mirror but the vertical range is sampled in the same way. By convention, the greater the range from the camera, the greater the corresponding depth value. The depth value at each sampled point, or pixel, is stored in a Human Information Processing Software (HIPS) format image file.

Figure 5–1 shows the range data obtained from a human face. The depth at each pixel is rendered in false colour with those points on the surface nearest the viewer (camera) represented in white. Those points further away are rendered in red through green to blue with dark blue representing those points furthest from the viewer.

The distance the object is moved in the i direction for each step is controlled by the operator and so the distance between, or the resolution of, the stripes is accurately known. In the equipment used here a maximum resolution of $10\mu m$ is possible. The distance between samples along the stripe, down to a minimum of $600\mu m$, and the resolution of the depth values, to a minimum of $100\mu m$, at each sample point are also controlled by the operator. The number of samples along either the i or j axis can be varied to produce images of different dimensions. The images used here are either 128 by 128 pixels or 256 by 256 pixels, with the same resolution in both dimensions.

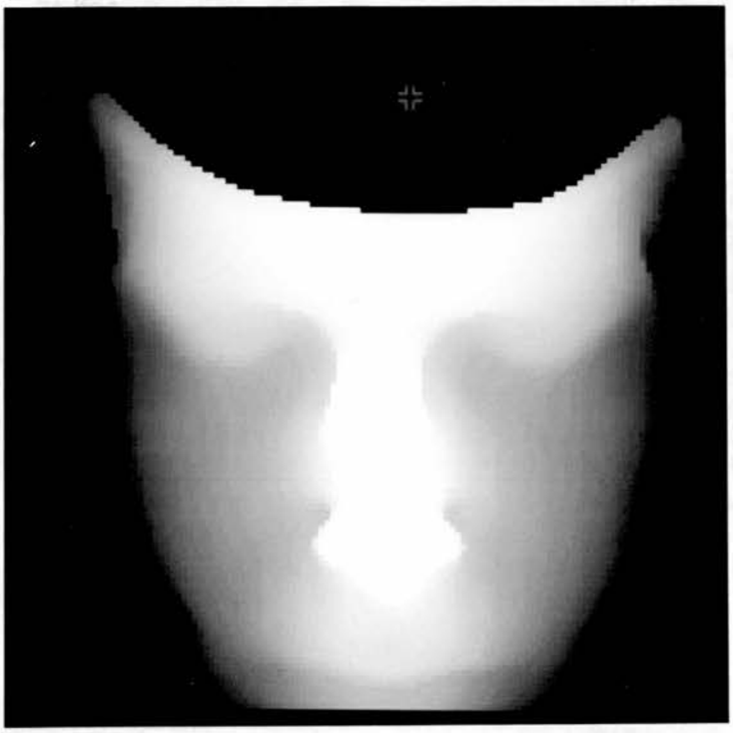


Figure 5–1: The Range Data Image of a Face

The resolution parameters for the i and j axes and the depth are stored in an image parameter file along with the projection of the camera system's optical axis in the image and the orientation of the camera in the world coordinate system. The latter enables points in the image coordinate system to be readily converted to x, y and z positions in the world coordinate frame. For convenience, here it is assumed that the x and y axes of the world coordinate frame are the same as the i and j axes of the image and so the depth values can be represented as distances along the z axis. A left–handed axis system is used so that greater depth values correspond, in this case, to greater values in the z direction.

The images produced are orthographically projected range images with the value at each sampled point on the object's surface representing its depth rather than its intensity. Representing the depth z of a surface as a function of the position of the sampled point $z = f(x, y)$ is known as Monge's form.

5.3 Object Surface Segmentation

Once the raw object surface data has been obtained, the next stage is to segment it into regions of uniform curvature. Existing software [Cai, 1990] is used to smooth the three dimensional image data and establish the curvature at each pixel in the image. Each point is labelled according to the sign of the Gaussian and mean curvatures and points with identical labelling are "grown" into regions. On the basis of the signs of the mean and Gaussian curvatures each region is assigned a shape as summarised below:

- PLANAR – The mean and Gaussian curvatures are both zero.
- POSITIVE CYLINDRICAL – The mean curvature is positive and the Gaussian curvature is zero.
- NEGATIVE CYLINDRICAL – The mean curvature is negative and the Gaussian curvature is zero.
- POSITIVE ELLIPTICAL – The mean curvature and Gaussian curvatures are both positive.
- NEGATIVE ELLIPTICAL – The mean curvature is negative and the Gaussian curvature is positive.
- HYPERBOLIC – The Gaussian curvature is negative.

In the original software [Cai, 1990] the hyperbolic surfaces were subdivided into SADDLERIDGE, MINIMUM and SADDLEVALLEY according to the relative magnitudes of the principal curvatures κ_α and κ_β. In the work described here the relative magnitudes of the curvatures of a surface are affected by deformation whereas their signs of curvature are not. Classifying all such surfaces as hyperbolic gives a more stable representation.

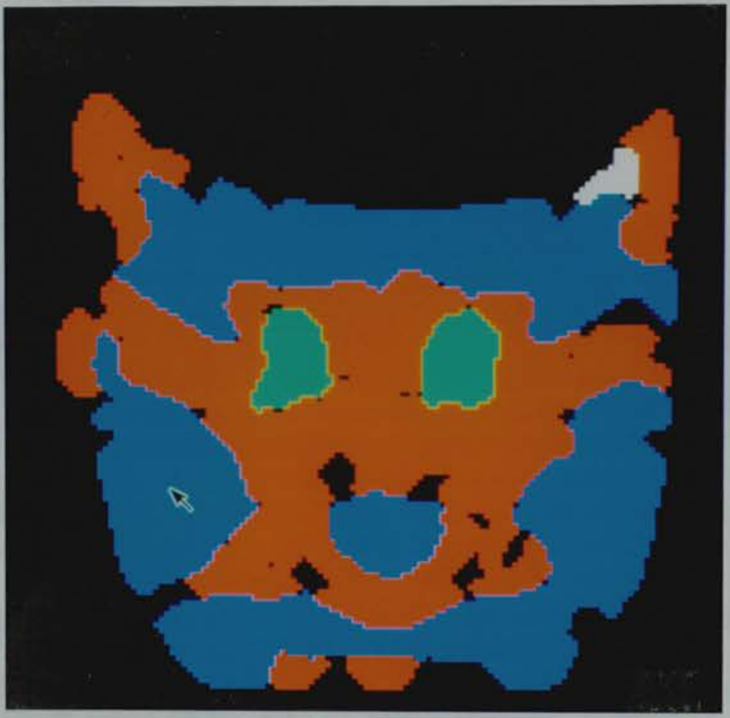


Figure 5–2: The Segmented Image of the Face

Once the regions have been identified they, and the pixels they contain, are placed in a list. Figure 5–2 shows the segmented image of the face. The convention used here and in all other range images depicted in this thesis is that planar surfaces are represented in yellow, positive cylindrical in orange, negative cylindrical in purple, positive elliptical in blue, negative elliptical in green and hyperbolic surfaces in red. Regions that are too small for their shape to be calculated are represented in white.

5.4 Object Surface Parameterisation

Having segmented the object surface data into smaller surface regions, the next stage is to estimate the parameters of each region.

The raw range data within each segmented surface in the list is smoothed in the x and y directions to reduce the effects of any noise. The new depth value z' for a point (x, y) in the range data smoothed in the x direction is obtained by local averaging:

$$z' = \frac{f(x-1,y) + 4f(x,y) + f(x+1,y)}{6} \tag{5.1}$$

A similar expression is used to smooth the range data in the y direction.

The range data can be alternately smoothed in the x and y directions until the effects of noise have been reduced sufficiently. Eight smoothing operations in each direction are generally sufficient.

The first–order partial derivatives of z' are p and q where:

$$p = \frac{\partial z'}{\partial x} \tag{5.2}$$

$$q = \frac{\partial z'}{\partial y} \tag{5.3}$$

are determined at each point in the range image by using the nearest neighbour difference functions:

$$p(x,y) = \frac{(f(x+1,y) - f(x-1,y))}{2x_{res}} \tag{5.4}$$

$$q(x,y) = \frac{(f(x,y+1) - f(x,y-1))}{2y_{res}} \tag{5.5}$$

The constants x_{res} and y_{res} are the camera resolution constants (the distance between samples) in the x and y directions respectively.

The second–order partial derivatives of z' are r, s and t where:

$$r = \frac{\partial^2 z'}{\partial x^2} \tag{5.6}$$

$$s = \frac{\partial^2 z'}{\partial x \partial y} \tag{5.7}$$

$$t = \frac{\partial^2 z'}{\partial y^2} \tag{5.8}$$

are determined by repeated use of the difference functions on the range image.

Unfortunately, each application of the difference function loses one pixel around the boundary of the surface. Where the segmented surface is small there may be too few pixels to determine the partial derivatives and it will not be possible to determine the parameters of the surface.

Using the partial derivatives obtained earlier then, as the surface is represented in Monge's form, the quantities E, F and G, known as the fundamental magnitudes of the first–order [Weatherburn, 1931], can be obtained at each point for which the first–order partial derivatives of the surface are available from the expressions:

$$E = 1 + p^2 \tag{5.9}$$

$$F = pq \tag{5.10}$$

$$G = 1 + q^2 \tag{5.11}$$

The quantities R, L, M and N, known as the fundamental magnitudes of the second–order, are obtained at all points on the surface where the second–order partial derivatives are available from the expressions:

$$R = \sqrt[2]{1 + p^2 + q^2} \tag{5.12}$$

$$L = \frac{r}{R} \tag{5.13}$$

$$M = \frac{s}{R} \tag{5.14}$$

$$N = \frac{t}{R} \tag{5.15}$$

Each point on the surface has two orthogonal principal curvatures κ_α and κ_β. The two values of κ can be found by solving the quadratic equation:

$$R^2\kappa^2 - (EN - 2FM + GL)\kappa + (LN - M^2) = 0 \tag{5.16}$$

The Gaussian curvature K is the product of the two principal curvatures:

$$K = \kappa_\alpha \kappa_\beta \tag{5.17}$$

The Gaussian curvature may either be found from the roots of equation (5.16) or, since this involves finding the roots of a quadratic at each point in the image, it can also be obtained from the equation:

$$K = \frac{LN - M^2}{R^2} \tag{5.18}$$

To obtain an estimate of the Gaussian curvature for the surface region as a whole K_{region}, the mean average of the Gaussian curvature at each point K_i in the region is used:

$$K_{region} = \frac{1}{n}\sum_{i=1}^{n} K_i \tag{5.19}$$

where there are n points in the surface that the Gaussian curvature can be calculated.

The mean curvature H is the average of the two principal curvatures:

$$H = \frac{\kappa_\alpha + \kappa_\beta}{2} \tag{5.20}$$

This may either be found from the roots of equation (5.16) or alternatively from the equation:

$$H = \frac{EN - 2FM + GL}{2R^2} \tag{5.21}$$

The mean curvature for the region as a whole H_{region} is found as above.

If the principal curvatures are known, the direction of the principal curvature κ_α at a point is given by the equation:

$$\theta = tan^{-1}\left(-\frac{\kappa_\alpha F - M}{\kappa_\alpha E - L}\right) \tag{5.22}$$

In the convention used here, θ is the angle between the direction of the principal curvature κ_α and the x axis. Since the principal curvatures are orthogonal, the direction of the principal curvature κ_β is $\theta + \frac{\pi}{2}$.

At each point on the surface, the components of the surface normal n are given by the expression:

$$\mathrm{n} = \frac{1}{R}(p, q, -1) \tag{5.23}$$

In the signing convention used here, a negative z component to the vector indicates that the surface normal is *towards* the viewer.

Because the projection of the object surface data in the image is orthographic, the area of the surface A_i represented by an individual point is given by the expression:

$$A_i = R_i x_{res} y_{res} \tag{5.24}$$

where, as before, x_{res} and y_{res} are the distances between sample points in the image in the x and y directions respectively. The area of the entire region is given by the sum of the areas represented by its individual points.

$$A_{region} = x_{res} y_{res} \sum_{i=1}^{n} R_i \tag{5.25}$$

where there are n points in the surface for which the area is to be calculated.

The centroid of the surface in image coordinates as projected in the image is estimated by an averaging process. The origin of the image coordinate system is the projection in the image of the optical axis.

The centroid of the surface in three dimensional coordinates is estimated using the centroid in terms of image coordinates and the image parameters.

5.5 Examples

As a test the half–cylinder in Figure 5–3 was passed under the laser striper, segmented, and its surface parameters calculated. The cylinder, which has a radius of $30mm$ and a length of $100mm$, was positioned such that the axes of principal curvature were aligned with the axes of the image and the centre of the object lay on the optical axis – in this case the centre of the image.

The following results were obtained:

```
REGION NUMBER 1
This region centred on image coordinates 64 64
Number of points : 4157
```

```
This region centred on 3-D (world) coordinates
POINT in frame world  (0.000, 0.000, 969.250)
Average surface normal is
VECTOR in frame world  (0.015, -0.000, -1.000)
Shape is            : HYPERBOLIC
First curvature     :   0.032712 (Radius :     30.57mm)
Second curvature    :  -0.000572 (Radius : -1748.92mm)
First direction     :   0.001225 radians (  0.07 degrees)
Second direction    :   1.572022 radians ( 90.07 degrees)
Mean curvature      :   0.016070
Gaussian curvature :  -0.000019
Surface area        : 4637.9mm^2
```

Although the quantities shown here are quoted to six decimal places the actual accuracy is a great deal less than this. By comparing the range data obtained with that of a perfect half–cylinder at the same position and orientation, the root mean squared error in the radius of the cylinder was estimated to be $\pm 0.5mm$.

The image coordinates represent the position of the centroid of the region in the HIPS image coordinate system.

The number of points are the number of pixels remaining in the region *after* performing the local differencing operations.

The three dimensional coordinates represent the position of the centroid of the surface in the world coordinate frame. Since the image resolution parameters are known, the positions are absolute distances, in this case in millimetres.

The mean and Gaussian curvatures for the region are the mean average for the points in the region. The principal curvatures for the region are derived from these.

The position, normal vector, curvature and axes of curvature are calculated correctly. The small differences in the normal vector, curvature and axes of curvature are caused by noise in the original range data. The second curvature of the cylinder is incorrect and should be zero. This is also due to the effect of noise in the original

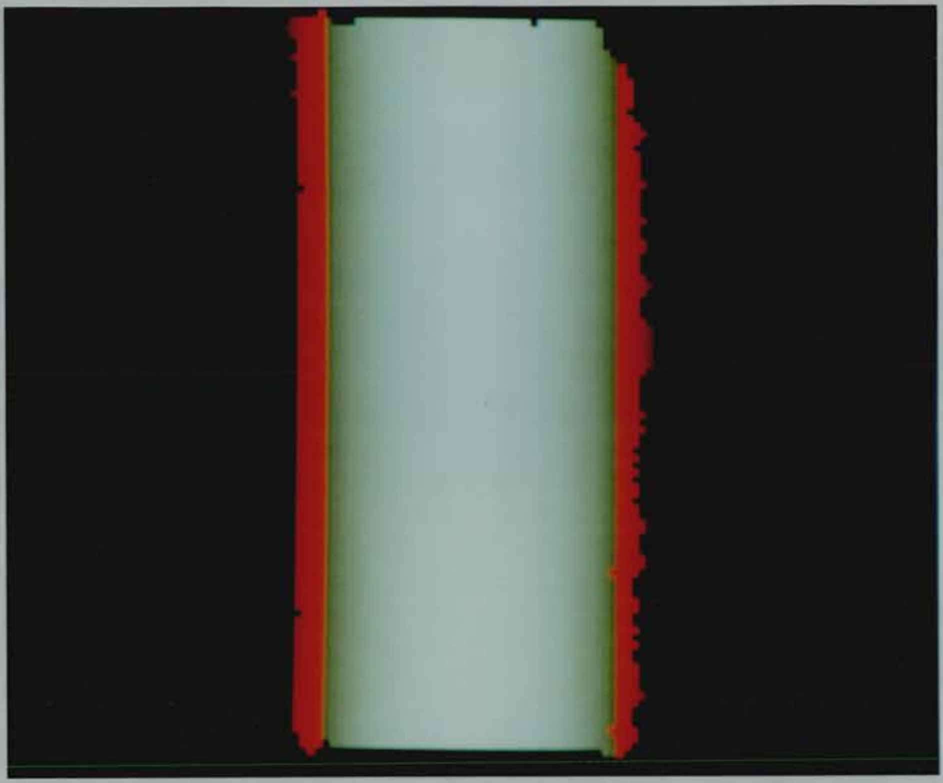


Figure 5–3: A Range Data Image of the Test Object

range data causing the curvature to become non–zero and the shape to be incorrectly classified as hyperbolic.

The area of the half–cylinder is too small, $4638mm^2$ instead of the actual figure of $9424mm^2$. This is due to the loss of the points around the edges of the segmented surface during the segmentation process and the calculation of the difference functions p, q, r, s and t. With one sample point for every square millimetre there were approximately 6000 points on the surface. After segmentation only 4723 points remained corresponding to a surface of approximately 89 by 53 pixels. The difference functions lost a further two points from around the edges of the surface giving a region of 85 by 49 pixels, a total of 4165 sample points, close to the actual figure of 4157.

The position of the half–cylinder was changed so that it was translated a few millimetres relative to the optical axis and the axes of curvature were rotated $\pi/4$ relative to the axes of the image.

The following results were obtained:

```
REGION NUMBER 2
This region centred on image coordinates 63 62
Number of points : 4795
This region centred on 3-D (world) coordinates
POINT in frame world  (-1.000, 2.000, 969.250)
Average surface normal is
VECTOR in frame world  (0.003, 0.003, -1.000)
Shape is             : HYPERBOLIC
First curvature      :   0.032913 (Radius :     30.38mm)
Second curvature     :  -0.001189 (Radius :   -841.34mm)
First direction      :   0.724253 radians ( 41.50 degrees)
Second direction     :   2.295050 radians (131.50 degrees)
Mean curvature       :   0.015862
Gaussian curvature   :  -0.000039
Surface area         : 5636.3mm^2
```

As before the position, normal vector and radius of the cylinder were calculated correctly within the limits of error. The second curvature of the cylinder shows a significant change from that calculated in the previous test image showing its susceptibility to any noise in the range data. The axes of curvature are not quite as expected. This is again due to the effects of noise. The calculated value for the area is smaller than expected for the reasons given previously.

The results given here were obtained from the segmented range data image shown in Figure 5–2. The results below correspond to the region shown in blue indicated by the cursor towards the lower left of the image.

```
REGION NUMBER 3
This region centred on image coordinates -21 -36
Number of points : 474
This region centred on 3-D (world) coordinates :
POINT in frame world  (-23.667, -40.572, 49.680)
Average surface normal is :
```

```
VECTOR in frame world  (-0.694, -0.177, -0.698)
Surface shape        :  POSITIVE ELLIPTICAL
First curvature      :  0.026576 (Radius :     37.62mm)
Second curvature     :  0.005292 (Radius :    188.97mm)
First direction      :  0.216706 radians ( 12.42 degrees)
Second direction     :  1.787502 radians (102.42 degrees)
Mean curvature       :  0.015934
Gaussian curvature :  0.000186
Surface area         :  943.5mm^2
```

Where a surface region contains only a few points, there may be too few to evaluate the difference functions and calculate the surface parameters accurately. In this case, the patch is assumed to be of planar shape and the parameters set accordingly. The centroid of the surface can be estimated approximately from the few pixels that were originally available.

The region below corresponds to the small region shown in white towards the upper right of the image in Figure 5–2.

```
REGION NUMBER 26
This region centred on image coordinates -27 21
Number of points 0
This region centred on 3-D (world) coordinates:
POINT in frame world  (-30.429, 23.667, 37.164)
Average surface normal is
VECTOR in frame world  (0.000, 0.000, 0.000)
Region 26 is too small, ASSUMED PLANAR
Surface shape        :  PLANAR
First curvature      :  0.000000 (Radius : Infinite)
Second curvature     :  0.000000 (Radius : Infinite)
First direction      :  0.000000 radians (  0.00 degrees)
Second direction     :  1.570796 radians ( 90.00 degrees)
Mean curvature       :  0.000000
```

```
Gaussian curvature :  0.000000
Surface area       :  0.0mm^2
```

Given a segmented image, the time taken to find the surface parameters is a function of the number of surfaces in the image and the total number of points in each. On a SUN SPARC 1+ microcomputer the process takes between five and eight seconds.

5.6 Conclusion

In this chapter the method is described whereby a description of an object's surface in terms of three dimensional data is obtained and processed prior to use by the recognition process.

The process of smoothing and segmenting the raw surface data into regions of uniform curvature is performed by the repeated use of nearest neighbour operations. This ensures that the computation is both fast and efficient and readily lends itself to implementation on locally connected parallel architecture machines such as the Cellular Logic Image Processor (CLIP) or the Distributed Array Processor (DAP).

The parameterisation of the surface regions is also carried out using nearest neighbour operations. The initial representation of the surface in terms of fundamental forms, although computationally expensive, means that all the required surface parameters, such as the curvatures, surface normals and area, can be calculated quickly from these.

Chapter 6

Model Selection

6.1 Introduction

Object identification is achieved by matching the features of an object with the features of a model. If the match is sufficiently good according to some appropriate criteria then the identity of the object is assumed to be that represented by the model. However a recognition system may contain many models in its model database and exhaustively matching the features of each with each feature in the image is at best time consuming and often impractical due to memory limitations. It would be convenient if those models representing only the objects in the image could somehow be identified in advance and selected from the database for rigorous feature matching.

Here nothing is initially known about the possible numbers and identities of the objects in the image and so the appropriate models cannot be selected immediately. The aim of the model selection process is to examine the object features in the image and quickly establish plausible object to model correspondences. Based on these correspondences, only those models for which there is sufficient evidence of a correspondence with an object in the image are selected for further matching.

Because the models are generic models and the objects in the image may differ greatly from the model it is difficult to determine the degree to which a single object feature fits that of a model. This is overcome by using combinations of object features.

If the features belong to the same object they must be subject to the same deformation which enables the deformation of the object to be recovered and the degree of fit estimated.

The problem approached here is *twofold*. Firstly there is the *efficient* selection of the correct models for an image which may contain many objects of unknown identity. Secondly there is the *estimation* of the deformation parameters for each object in the image. The solutions to these two aspects of the problem are linked in that accurate model selection requires the deformation to be known and the identity of the appropriate model must be hypothesised for the deformation to be calculated.

The models described here have a hierarchical structure. At the lowest level configurations of surface primitives form parts while at higher levels model parts and larger surface primitives form superparts or the model itself. The model selection process falls into two sections. Initially the part selection process selects model parts or superparts based on matches between object and model surfaces. In the later stages the superpart selection process selects model superparts or models themselves based on correspondences between object parts and model parts. This division is made because the generation of interpretations, the constraints used to eliminate incorrect interpretations and the evidence used in making a selection are very different, because of the increased and more reliable information available from a combination of parts rather than a combination of surfaces.

The process of model selection, as described here, is compared with previous research on the use of data driven recognition systems, and the differences discussed.

6.2 Model Part Selection

The aim of the part selection process is to use correspondences between object and model surfaces to establish which model parts or superparts are most likely to represent the object parts in the image. Although both parts and superparts that contain surfaces can be selected by this process, to avoid confusion only the more common case of selecting a part consisting entirely of surfaces will be discussed. The process of selecting a superpart by plausible surface correspondences is the same.

6.2.1 Theory

The part selection process consists of four stages. Initially there is the generation of combinations of object surfaces. These object surfaces are then interpreted with the aid of a model data base. The plausibility of each interpretation is considered and finally the number of interpretations are reduced by finding those that are inconsistent with a set of simple constraints.

Forming Combinations of Image Surfaces

Generating combinations of object surfaces and matching these with model surfaces enables the deformation of the object relative to the model to be established.

If there are I surfaces in the image and all combinations of n (n–tuples) are to be generated then the number of possible combinations is given by:

$$number\ of\ combinations\ =\frac{I!}{(I-n)!n!} \tag{6.1}$$

Obviously since I is fixed for a given image the number of combinations to be generated is dependent on n. If n is small then the number of combinations to be matched is small but the amount of evidence available to recover the deformation parameters and determine a good correspondence is also small. A larger value for n will give a better estimate of any deformation and a better determination of fit but usually generates so many combinations that matching would be impractical. For large values of n, where $n \rightarrow I$, the amount of evidence available is very large and the number of combinations generated *decreases*. However, as stated earlier, an image contains an unknown number of objects and if n is very large, it is unlikely that any of the n–tuples generated would consist entirely of the surfaces of one object. The value of n should therefore be as small as possible while still ensuring that correct object to model correspondences can be identified. If the value of n is small and the value of I

is very large then Equation (6.1) becomes:

$$number\ of\ combinations \approx I^{n} \tag{6.2}$$

Interpreting Combinations of Image Surfaces

Having generated all n–tuples of the surfaces in the image, these must be matched with the surface primitives of each model part. If it is assumed that the object surfaces in a combination belong to the same object part then they only need be matched with the surfaces within each model part in turn. If a model part p consists of m_p surfaces then the number of ways that a combination of n object surfaces can be matched with the model surfaces is m_p^n. This assumes that a model surface can be matched to several object surfaces at the same time. If each model surface must match a different object surface, the number of possible matches of a given combination of object surfaces becomes the number of permutations of the model surfaces:

$$number\ of\ permutations\ = \frac{m_p!}{(m_p - n)!} \tag{6.3}$$

Each match of a combination of object surfaces with a permutation of model surfaces is an interpretation of the object surfaces. Combining the two equations gives the number of possible interpretations of the surfaces in the image using a single model part:

$$number\ of\ interpretations\ = \frac{I!}{(I-n)!n!}\frac{m_p!}{(m_p - n)!} \tag{6.4}$$

If there are M parts in the model database the total number of interpretations of the surfaces in the image becomes:

$$total\ number\ of\ interpretations\ = \frac{I!}{(I-n)!n!}\sum_{p=1}^{M}\frac{m_p!}{(m_p - n)!} \tag{6.5}$$

Provided that the n object surfaces in the combination belong to the same object part and that the object part is represented by a model part in the model database which contains corresponding surfaces, then the correct interpretation of that combination, and any other combinations that meet these criteria, will be generated.

An equally valid approach would be to generate permutations of image surfaces and to match these with combinations of model part surfaces. Equation (6.5) becomes:

$$number\ of\ interpretations\ = \frac{I!}{(I-n)!}\sum_{p=1}^{M}\frac{m_p!}{(m_p-n)!n!} \qquad (6.6)$$

The number of interpretations is the same as before and a correct interpretation, if one exists, will be generated for each permutation of image surfaces.

Establishing the Plausibility of an Interpretation

As possible matches for each object surface in a combination are made, more information about the object and its deformation parameters in that interpretation becomes available. A match between a single object surface and a single model surface can suggest some information as to the likelihood of a correspondence but no information about the object as a whole. A match between two object and two model surfaces gives more information about the plausibility of a correspondence and furthermore enables a rough estimate of the scale and stretch to be determined along the line that joins the surfaces. Where three surfaces are matched the parameters of global scale and part stretch, as defined in Chapter 4, can usually be determined.

Once the deformation parameters have been established, the model part can be deformed to fit the configuration of the object surfaces. The degree to which the surfaces of the deformed model part fit those of the object part, the "degree of fit", gives a measure of the likelihood of a correct interpretation.

Reducing the Number of Interpretations

The number of possible interpretations will usually be very large, often too large to be practical, and the calculation of the deformation parameters and degree of fit for each interpretation will be time consuming. If incorrect pairings could be identified quickly then the number of interpretations would be reduced and only the parameters of the correct interpretations need be determined. The shape of a surface, the distance between two surfaces, the configuration of a number of surfaces and the scale and stretch parameters can all provide limits, or constraints, on the matches allowed within an interpretation. Where a match fails to meet the constraints imposed, that interpretation is invalid and can be discarded. As matches are found for each surface in the combination, the earlier an interpretation can be identified as invalid the greater the gain in saved time and memory.

6.2.2 Implementation

The implementation of the model part selection process must address the four issues described previously. This requires the *efficient* generation and interpretation of combinations of object surfaces.

Generating Combinations of Image Surfaces

The first stage of the part selection process generates all possible combinations of *three* object surfaces. Combinations of three object surfaces are chosen as three object to model correspondences are the minimum number normally required to estimate the combined scale and stretch parameters S_{gx}, S_{gy} and S_{gz}. A three surface correspondence is also the minimum required by the part verification process to establish a coordinate frame transformation. The *maximum* number of combinations of three surfaces (or three–tuples) is given by Equation (6.1) but in practise the actual number of combinations generated is often less than this figure. Usually the image will contain several fragmentary surfaces that, as described in Chapter 5, are too small for any surface parameters to be obtained. These surfaces patches contain too little information to be of any use in the selection process and are ignored for the time being.

Interpreting a Combination Using an Interpretation Tree

The next stage of the process seeks to generate all possible interpretations of each generated combination of object surfaces by matching each combination with each permutation of model surfaces for each model part in turn. For this purpose an interpretation tree, as proposed by Gaston and Lozano–Pérez [Gaston & Lozano-Pérez, 1984], is used. In their approach an interpretation tree was generated for each model to represent all possible matches between the model features and the image data. Here an interpretation tree is generated each time a combination of object surfaces is matched with a model part. This results in a greater number of trees but each has a depth of only three reducing the complexity. A further difference is that during part selection a model surface may only be matched to one object surface. This means that the number of branches in the tree is thinned at the lower nodes. Figure (6–1) shows the tree generated to match the three object surfaces in a combination with a single model part of four surfaces. Where a model part contains fewer than three surfaces it cannot be used in the selection process. Recognition of parts with less than three surfaces is covered in the model verification chapter. To save time the model database contains a list of all model parts, superparts and models with three or more surfaces. The selection process only need examine these to generate possible interpretations.

Estimating the Deformation Parameters in an Interpretation

Once two of the object surfaces have been matched with two of the model surfaces, a rough estimate of the combined scale and stretch can be determined. If the centres of the two object surfaces are separated by a distance D in the world coordinate frame, see Figure (6–2), and the corresponding model surfaces are separated by a distance d in the model part coordinate frame, see Figure (6–3), and if the individual variations in surface position are assumed to be negligible, then the combined effect of the global scale and part stretch S_{gxyz} along the line joining the two surfaces is approximated by:

$$S_{gxyz} \approx \frac{D}{d} \tag{6.7}$$

Once correspondences have been obtained between three object surfaces and three model surfaces then it is possible to make an initial estimate for the combined global scale and part stretch parameters S_{gx}, S_{gy} and S_{gz}. If the positions of the three model surfaces in the model coordinate frame are represented by (x_i, y_i, z_i) then the distance between any two surfaces j and k in the model coordinate frame, see Figure (6–4), is given by:

$$distance = \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2 + (z_j - z_k)^2} \tag{6.8}$$

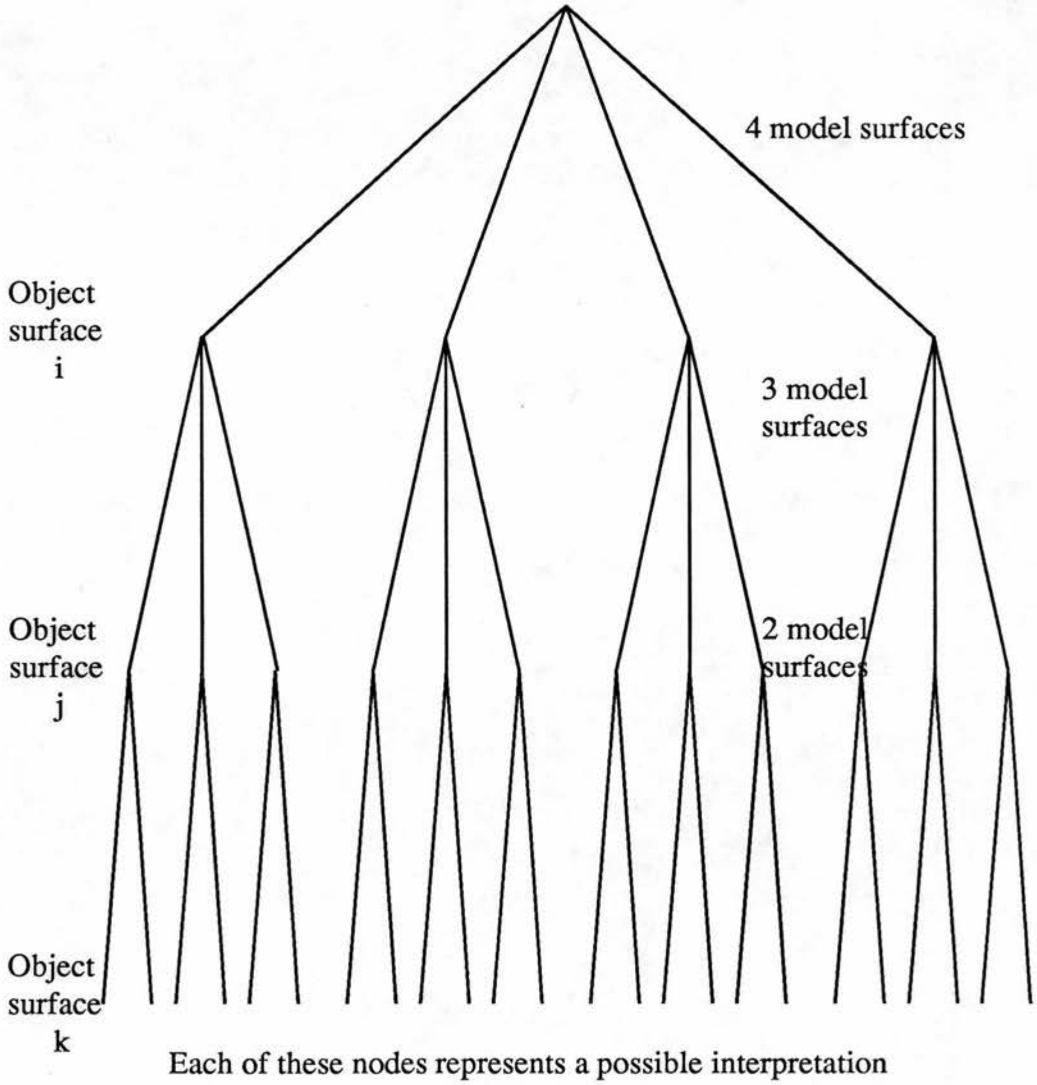


Figure 6–1: An Interpretation Tree for Three Object Surfaces and a Model Part of Four Surface Primitives

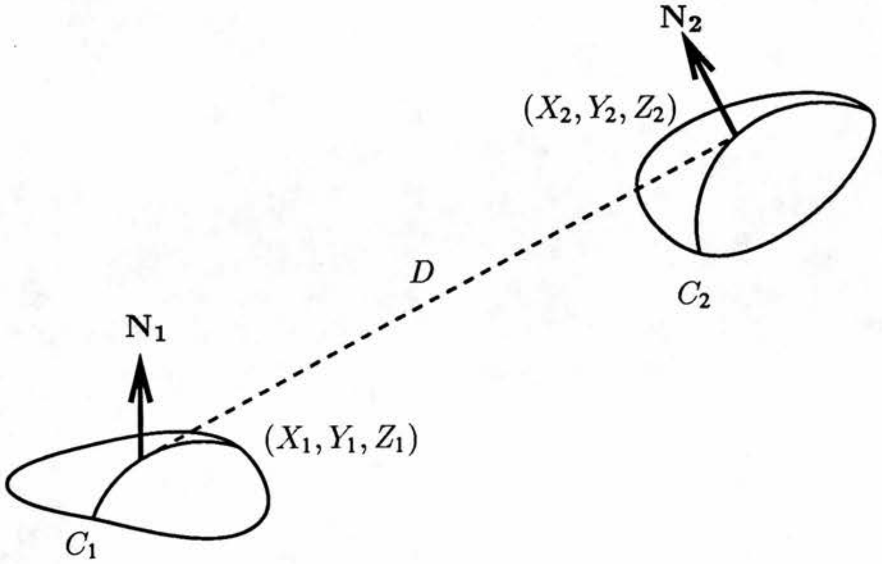


Figure 6–2: A Pair of Object Surfaces

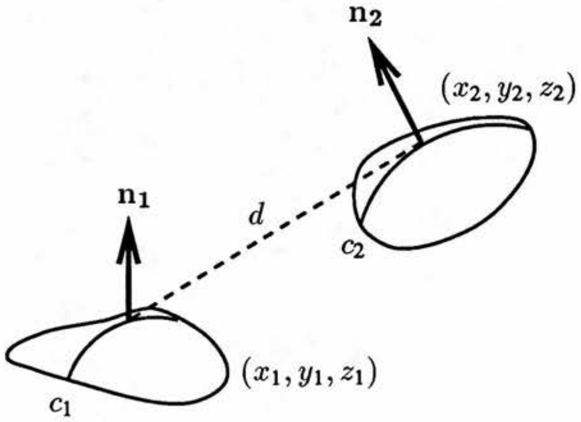


Figure 6–3: The Corresponding Pair of Model Surfaces

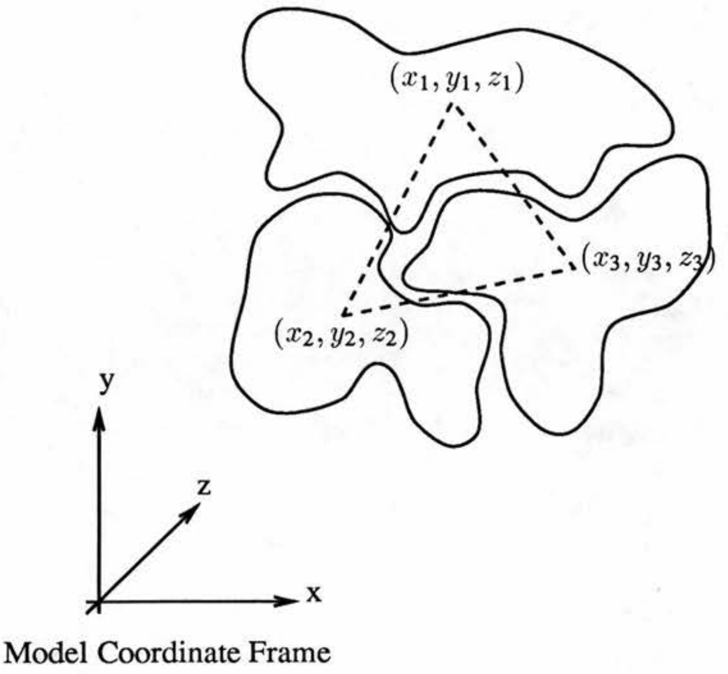


Figure 6–4: A Permutation of Three Model Surfaces

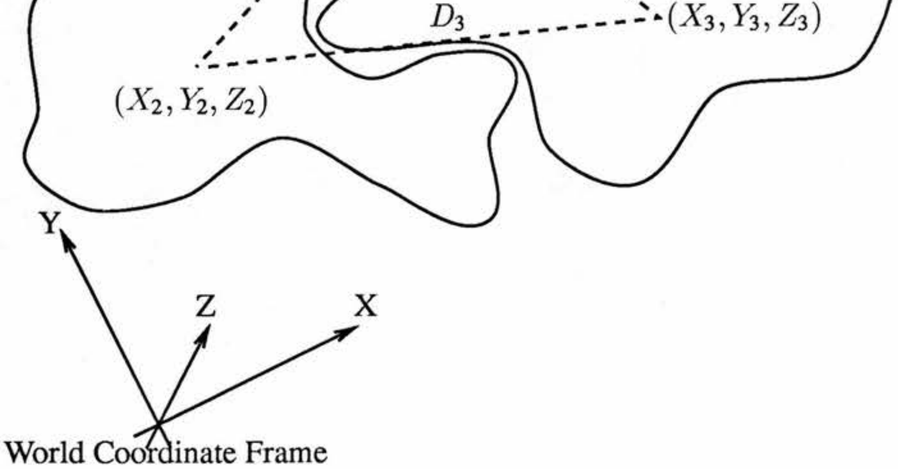


Figure 6–5: A Combination of Three Object Surfaces

If the model part is scaled and stretched, the distance between the surfaces in the model frame becomes:

$$stretched\,distance = \sqrt{(S_{gx}(x_j - x_k))^2 + (S_{gy}(y_j - y_k))^2 + (S_{gz}(z_j - z_k))^2} \tag{6.9}$$

The distance D_{JK} between the corresponding object surfaces J and K in the world coordinate frame, see Figure (6–5), may be found by applying Equation (6.8) to the positions of the object surfaces. Using the result with Equation (6.9) gives the equality:

$$(S_{gx}(x_j - x_k))^2 + (S_{gy}(y_j - y_k))^2 + (S_{gz}(z_j - z_k))^2 = D_{JK}^2 \tag{6.10}$$

Since there are three surfaces there are three sets of equations which can be solved for S_{gx}, S_{gy} and S_{gz} as detailed in Section 8.2.1.

There are several cases where it is not possible to uniquely determine all three components of scale and stretch and an alternative approach must be sought.

1. The first case occurs where the centres of the three *model* surfaces all lie upon the same line in the part coordinate frame. Fortunately, because this condition arises from the configuration of the surfaces in the model rather than in the image, it is possible to test for it during the construction of the model. A warning flag is set within the model part, to be loaded at run–time. Three model surfaces occur in this configuration so rarely that if the flag is set then the corresponding permutation of model surfaces can normally be discarded from the selection process without undue loss of information. The model part will then either be selected by a different combination of object surfaces or be searched for during superpart verification after selection of the superpart by a different part of the same object.

2. The second condition occurs when the positions of all three *model* surfaces form a triangle which is parallel to the $x = 0$, $y = 0$ or $z = 0$ planes in the model part coordinate frame. When this occurs the model will have no component along the appropriate axis and it is not possible to determine the scale and stretch along that axis. As before this problem is due to the configuration of the model surfaces and can be detected during construction of the model. Where it occurs

a flag is set within the model part which indicates that only two components of the scale and stretch are to be determined. Instead of a set of three equations and three unknowns it becomes an overspecified problem of three equations and two unknowns. By using a least squares approach, as covered in Section 8.4.1, the best estimates of these values are determined.

3. A third condition arises where the central points of the three model surfaces form a right–angled triangle in the model coordinate frame. Each of the three equations obtained using Equation (6.10) is essentially the squared length of one side of this triangle. In the case of a right–angled triangle, the sum of two of the three equations will be equal to the third. Therefore, with only two independent equations available, an exact solution for all three variables is not possible.

   The problem of model surfaces forming a right–angle triangle occurs sufficiently often, especially with man–made objects, that disregarding those permutations of model surfaces for which it occurs would lose too much useful information. Although each of the components of the combined scale and stretch can no longer be individually determined a *rough estimate* of the scale and stretch can still be achieved. The two independent equations, each with three unknowns, are taken and copied into three sets. In each of the three sets two of the coefficients are replaced by the same variable (i.e. they are assumed to be equal) reducing the problem to that of three sets of two equations each with two unknowns. Each equation set is solved giving three possible sets of solutions for the combined scale and stretch, each of which are possible but *none of which are necessarily accurate*. In resolving deformations the criterion stated previously is that where possible a deformation should be explained as far as possible by a change in global scale and the stretch of a part minimised. Therefore, of the three possible sets of solutions, that which shows the *smallest* difference between the two estimated stretch coefficients is the one selected.

   If there is *only* a change in global scale then the combined coefficients of scale and stretch will be evaluated correctly. More generally though stretch will be present and an accurate estimate of this stretch may not be made. This is a form of "aperture problem" where, without an accurate estimate of the scale and stretch, the model verification process will give an inaccurate estimate of an object's rotation and translation. Ideally other permutations of model surfaces will give a more accurate selection of the same part. However, as covered in the model verification chapter, the matching of further surfaces will enable any initial estimate to be refined.

4. The last, and by far the most common, condition occurs where the central points of two of the model surfaces are symmetric about a line parallel to either the x, y or z axes in the model part coordinate frame. When this occurs, two of the three equations become linearly dependent and, as in the previous case, a solution of the three variables with the two remaining independent equations is not possible. Moving the three model surfaces so that they were skewed relative to the axes of the coordinate frame could solve this problem but then raises a further one in that the stretch coefficients S_{gx}, S_{gy} and S_{gz} along each coordinate frame axis would not then relate directly to the "natural" axes of stretch of the part.

   The problem of symmetric model surfaces occurs sufficiently often that disregarding those configurations of model surfaces for which it occurs would eliminate too many potential selections. Although the combined scale and stretch can no longer be uniquely determined, as with the previous case an approximate estimate is still possible. The two offending model surfaces are symmetric about a line parallel to one of the axes in the model coordinate frame. They also lie in a plane to which the line of symmetry is a normal. The model coordinate frame is rotated so that the surfaces lie in a plane parallel to either of the model axes normal to the axis of symmetry. When this is done then, of the two remaining independent equations, one will have non–zero coefficients for only one of the directions of stretch orthogonal to the line of symmetry. An exact solution for this coefficient of scale and stretch is now possible. In the remaining equation there are now (normally) two unknown coefficients of stretch which could take on a range of values. Once again the criterion that minimises part stretch is used. The stretch of this part is assumed to be minimised when the remaining two unknown components of stretch are equal. This means that the remaining equation now contains only one unknown for which it can be solved.

Although one of the coefficients of scale and stretch will be correctly determined, the remaining two may not be accurate. This leads to the same aperture problem as in the previous case. The results included in both this chapter and the next include several examples of this case and illustrate its effect on the selection and verification processes as a whole.

Establishing the Degree of Fit Between the Deformed Model Part and the Object

To determine how well an interpretation explains the data in the image it is necessary to determine the degree of fit between the object surfaces and those of the deformed model part.

Given three surfaces i, j and k in the deformed model part the angle between any two surface normal vectors, for example $\mathbf{n}_i$ and $\mathbf{n}_j$, is resolved in terms of their relative angles of slant and tilt (see Section 8.2.2). These are compared with the angles of slant and tilt between the corresponding object surface normals $\mathbf{n}_I$ and $\mathbf{n}_J$ as a measure of fit.

The Gaussian curvature of each deformed model surface K_{di}, K_{dj} and K_{dk} can be determined (see Section 8–4). The curvatures of the deformed model surfaces can be compared directly with the Gaussian curvature of the corresponding object surfaces K_I, K_J and K_K to provide a further measure of fit.

The degree of fit function is a weighted average of these two measures given by:

$$\begin{aligned} degree\ of\ fit \;=\; & \alpha((slant(\mathbf{n}_i,\mathbf{n}_j) - slant(\mathbf{n}_I,\mathbf{n}_J))^2 + \\ & (slant(\mathbf{n}_i,\mathbf{n}_k) - slant(\mathbf{n}_I,\mathbf{n}_K))^2 + \\ & (slant(\mathbf{n}_j,\mathbf{n}_k) - slant(\mathbf{n}_J,\mathbf{n}_K))^2 + \\ & (tilt(\mathbf{n}_i,\mathbf{n}_j) - tilt(\mathbf{n}_I,\mathbf{n}_J))^2 + \\ & (tilt(\mathbf{n}_i,\mathbf{n}_k) - tilt(\mathbf{n}_I,\mathbf{n}_K))^2 + \\ & (tilt(\mathbf{n}_j,\mathbf{n}_k) - tilt(\mathbf{n}_J,\mathbf{n}_K))^2)/6 + \\ & \beta((K_{di} - K_I)^2 + \\ & (K_{dj} - K_J)^2 + \\ & (K_{dk} - K_K)^2)/3 \end{aligned} \tag{6.11}$$

where α and β are the weighting coefficients discussed in Section 3–11.

This metric is not particularly accurate as the calculation of the combined scale and stretch factors assumes that no individual variation is present in the position of each surface. In the presence of significant variation the estimates of the scale and stretch will be inaccurate, resulting in inaccurate predictions of the orientation and curvature of a deformed model surface. This could lead to an incorrect interpretation being given a better degree of fit than a correct one but since all valid interpretations are passed to the verification process this should affect only the efficiency of the recognition process.

Ranking the Interpretations

The interpretations are placed in one of several "league tables" and ranked according to their degree of fit. Which league table an interpretation is placed in depends on the level of the selected part or superpart in the model. Taking as an example the model of the human head, see Figure (3–6), the parts which form the lowest level of the model such as the eyes and nose would be placed in the first league table if selected. At the next level up in the model the head superpart consists of both surfaces and parts. If this superpart is selected on the basis of plausible surface correspondences then it is placed in the second league table. Each league table therefore ranks selected parts from the same level within the appropriate models. Since the features represented at a given level in a model are intended to be of the same size or level of detail then each league table can be considered as loosely grouping parts of similar size.

The selections in these league tables form the input to the model part verification process.

6.2.3 Reducing the Number of Interpretations by Pruning the Interpretation Tree

Although this approach guarantees to generate the correct interpretation of a combination of three object surfaces, if one exists and does not fall under case 1 above, the total number of interpretations generated will be large. An image might contain 30 surfaces and a model part might contain 5 surfaces. From Equation (6.4) this would yield a possible 243600 interpretations of the image data based on that one model part alone. The number of possible interpretations can be reduced by the use of constraints to identify those matches that are incorrect. Where an incorrect match is identified all interpretations of which it forms part can be eliminated enabling whole branches to be "pruned" from the interpretation tree.

The Shape Constraint

Changes in the global scale and stretch of a part cause the magnitudes of the surface curvatures to change. The signs of curvature and the surface shape, as defined in Section 5.3, remain the same. Individual variations in the curvature of a surface can change its shape but since these are by definition small, only those surfaces which have zero or near zero curvatures are affected.

The shape constraint therefore checks that the shape of the model surface matches that of the corresponding object surface according to the following set of rules:

- A planar model surface may match any shape of object surface. (Small variations in curvature could deform a plane into any shape.)
- A negative cylindrical model surface may match a negative cylindrical, a hyperbolic or a negative elliptical object surface. (Variation in the zero curvature of the cylinder could deform the cylindrical surface.)
- A positive cylindrical model surface may match a positive cylindrical, a hyperbolic or a positive elliptical object surface. (For the same reasons given above.)
- A hyperbolic model surface may only match a hyperbolic object surface.
- A negative elliptical model surface may only match a negative elliptical object surface.
- A positive elliptical model surface may only match a positive elliptical object surface.

The Two Dimensional Scale and Stretch Constraint

Where there is a correspondence between two object surfaces and two model surfaces, the distance between the object surfaces provides a further constraint. If the distance between the model surfaces is d and between the object surfaces is D then the ratio D/d must not fall outside the maximum and minimum limits of the combined effects of scale and stretch as specified in the model.

If the upper and lower limits on the three components of combined scale and stretch are S_{gx_u}, S_{gy_u}, S_{gz_u} and S_{gx_l}, S_{gy_l} and S_{gz_l} respectively then the upper limit of combined scale and stretch along a line, S_{gxyz_u}, is given by:

$$S_{gxyz_u} = \sqrt{S_{gx_u}^{2} + S_{gy_u}^{2} + S_{gz_u}^{2}} \tag{6.12}$$

The lower limit of the combined scale and stretch S_{gxyz_l} is given by:

$$S_{gxyz_l} = \ minimum\ of\ (S_{gx_l}, S_{gy_l}, S_{gz_l}) \tag{6.13}$$

The upper and lower limits are calculated and stored within each model part to be loaded at run–time.

The Triple Scalar Product Constraint

This constraint requires a correspondence between three object surfaces and three model surfaces. Although the deformation of a model part can alter the positions of the surfaces within it, if a line is drawn between two of the surfaces in the interpretation then the third surface will be on one or other side of this line and will remain so regardless of the deformation of the model.

If the three model surfaces are i, j and k then the surface normal vectors are $\mathbf{n}_i$, $\mathbf{n}_j$ and $\mathbf{n}_k$ and $\mathbf{v}_{ij}$ and $\mathbf{v}_{ik}$ are the *unit vectors* from the centre of surface i to surface j and surface i to surface k respectively. For a given model surface, in this case i, the triple scalar product q associated with it is given by:

$$q = \mathbf{n}_i.(\mathbf{v}_{ij} \times \mathbf{v}_{ik}) \tag{6.14}$$

If the *sign* of the triple product matches the sign of the triple product of the three object surfaces then the correspondence meets the constraint. The triple products associated with each permutation of surfaces within a model part are calculated during the construction of the model and stored within each model part.

The Three Dimensional Scale and Stretch Constraint

This constraint also requires a three surface correspondence between object and model. By evaluating the combined effects of global scale and stretch for the object part, these values can be checked against the limits imposed on the global scale and stretch in the corresponding model and model part.

The Relative Efficiency of the Three Constraints

The shape constraint is the most powerful in that it requires only one model and one object surface to be matched. This enables pruning to take place at the highest level of the interpretation tree where the largest number of nodes can be eliminated. The shape constraint can be applied at any level of the tree and as it is the easiest to apply is always tested for first. The two dimensional scale and stretch constraint allows pruning at the second level of the tree which is of lesser effectiveness. The triple product and three dimensional scale and stretch constraints eliminate interpretations at the lowest nodes in the tree. Although this will reduce the number of interpretations it is still necessary to generate these nodes before they can be eliminated. The three dimensional scale and stretch constraint is the most expensive to test for and so is evaluated last.

Identifying a Valid Interpretation

Those interpretations which meet the imposed constraints are regarded as valid interpretations of that combination of object surfaces. At this stage it is possible for a combination to have more than one interpretation though all but one, or possibly all, will be incorrect. It is also possible for an object surface to occur in two or more combinations all of which have valid interpretations. This can happen where both the object and model parts contain more than three surfaces. In this case, different three–tuples of the object surfaces can be correctly interpreted as corresponding to different permutations of the surfaces in the *same* model part. Although each such interpretation is correct, the deformation parameters determined in each will usually differ slightly due to the effects of variation. Where the same surface occurs in two or more combinations which have valid interpretations corresponding to *different* model parts, only one, or possibly none, can be correct.

6.2.4 Results

There are three sets of results in this section. The first set were obtained from a range of synthetically generated images designed to test the different aspects of the model part selection process in isolation. The second set of results show the application of the process to the images obtained from a class of real, but very simple, objects. These objects consist of only one visible part and the deformation is limited to a single axis. The final set of results were obtained from a class of complex shapes consisting of several parts and exhibiting a complicated structure of deformation.

The Class of Synthetic Faces

To test the various aspects of the part selection process a range of images were *synthetically generated.* The basic object, an idealised human face, that appears in each image is represented *exactly* by the corresponding model. In all but the first image this basic object is altered to test one aspect of the part selection process in isolation. The results of the part selection process for these images are summarised in Tables 6–1, 6–2 and 6–3.

The first image contains the basic synthetic face object (see Figure A–1). There is no rotation, translation (except along the Z–axis), scaling, stretching or variation of the object relative to the model. The synthetic face model consists of four surfaces – the forehead, two eyes and a chin and a part – the synthetic nose. The synthetic nose part consists of three surfaces, the nose ridge and the left and right nose hollows. In both model and object therefore, there are seven surfaces.

The process starts by generating all combinations of three object surfaces of which, from Equation (6.1), there are 35. Each of these combinations is compared with each

RESULTS FOR THE CLASS OF SYNTHETIC FACES					
Image	Surfaces in image	Total no. interps	No. of object parts(> 3surf)	No. model selections	No. correct selections
Standard Face	7	6510	2	21	5
Rotated Face	7	6510	2	21	5
Translated Face	7	6510	2	21	5
Scaled Face	7	6510	2	18	5
Stretched Face	7	6510	2	24	5
Face with variation	7	6510	2	21	5

Table 6–1: Summary of the Selections Made During Model Part Selection

RESULTS FOR THE CLASS OF SYNTHETIC FACES						
Image	S_{gx}	S_{gx}	S_{gy}	S_{gy}	S_{gz}	S_{gz}
	True	Estim.	True	Estim.	True	Estim.
STANDARD FACE						
Nose part	1.000	1.000	1.000	1.000	1.000	1.000
Face model	1.000	1.000	1.000	1.000	1.000	1.000
Face model	1.000	1.000	1.000	1.000	1.000	1.000
Face model	1.000	1.000	1.000	1.000	1.000	1.000
Face model	1.000	1.000	1.000	1.000	1.000	1.000
ROTATED FACE						
Nose part	1.000	1.000	1.000	1.000	1.000	1.000
Face model	1.000	1.000	1.000	1.000	1.000	1.000
Face model	1.000	1.000	1.000	1.000	1.000	1.000
Face model	1.000	1.000	1.000	1.000	1.000	1.000
Face model	1.000	1.000	1.000	1.000	1.000	1.000
TRANSLATED FACE						
Nose part	1.000	1.000	1.000	1.000	1.000	1.000
Face model	1.000	1.000	1.000	1.000	1.000	1.000
Face model	1.000	1.000	1.000	1.000	1.000	1.000
Face model	1.000	1.000	1.000	1.000	1.000	1.000
Face model	1.000	1.000	1.000	1.000	1.000	1.000

Table 6–2: The Deformation Parameters Estimated During Model PartSelection

RESULTS FOR THE CLASS OF SYNTHETIC FACES						
Image	S_{gx}	S_{gx}	S_{gy}	S_{gy}	S_{gz}	S_{gz}
	True	Estim.	True	Estim.	True	Estim.
SCALED FACE						
Nose part	2.000	2.000	2.000	2.000	2.000	2.000
Face model	2.000	2.000	2.000	2.000	2.000	2.000
Face model	2.000	2.000	2.000	2.000	2.000	2.000
Face model	2.000	2.000	2.000	2.000	2.000	2.000
Face model	2.000	2.000	2.000	2.000	2.000	2.000
STRETCHED FACE						
Nose part	1.000	1.000	1.100	1.078	1.000	1.078
Face model	1.000	1.000	0.800	0.824	1.000	0.824
Face model	1.000	1.000	0.800	0.800	1.000	1.000
Face model	1.000	1.000	0.800	0.800	1.000	1.000
Face model	1.000	1.000	0.800	0.817	1.000	0.817
FACE WITH VARIATION						
Nose part	1.000	1.000	1.000	1.000	1.000	1.000
Face model	1.000	1.000	1.000	1.000	1.000	1.000
Face model	1.000	1.000	1.000	1.013	1.000	1.013
Face model	1.000	0.976	1.000	1.009	1.000	1.107
Face model	1.000	0.976	1.000	1.009	1.000	1.107

Table 6–3: The Deformation Parameters Estimated During Model PartSelection

permutation of surfaces in each model part, superpart or model that contains three or more surfaces. There are currently seven such model parts in the database that, from Equation (6.3) yield 186 possible permutations and, from Equation (6.5), a maximum of 6510 interpretations. The application of the constraints described earlier enables the majority of the incorrect interpretations to be eliminated. After pruning, only 21 interpretations remain, a reduction by a factor of 310. Only five of the combinations in fact have valid interpretations and all of these are retained by the pruning process. There is a degree of redundancy in the selections in that while the synthetic nose part is correctly selected but once, the synthetic face model is correctly selected four times. This is due to the fact that all four of the surfaces represented by the model are visible in the image and so four combinations of three surfaces are generated, each of which makes the same selection.

In this case the model is an exact representation of the object so that there is no scaling, stretching or variation present. The combined scale and stretch parameters S_{gx}, S_{gy} and S_{gz} for each correct selection are calculated to be exactly 1.0, 1.0 and 1.0, the expected values, for both the synthetic face model and the synthetic nose part.

The second image contains the same basic synthetic face object as before but this time it has been rotated by $\pi/2$ (see Figure A–2). The same object surfaces are visible in the image and the maximum number of interpretations is again 6510. Since the constraints are coordinate frame independent, once again the possible interpretations are pruned to 21. These 21 interpretations are the same interpretations, with the same estimates of combined scale and stretch, as those obtained from the previous image. Since the determination of the parameters of deformation is also coordinate frame independent, the combined scale and stretch parameters S_{gx}, S_{gy} and S_{gz} for each correct selection are calculated to be exactly 1.0, 1.0 and 1.0.

The third image contains the same basic synthetic face object as the first but it has been translated by small distances in the X, Y and Z directions in the world coordinate frame (see Figure A–3). Once again the constraints reduce the number of interpretations to the same 21 and the combined scale and stretch parameters S_{gx}, S_{gy} and S_{gz} of the correct interpretations are calculated to be 1.0, 1.0 and 1.0.

The fourth image has the same synthetic face object but this time it has been uniformly scaled by a factor of two. This was achieved by changing the values in the camera parameter file, rather than the size of the object in the image (see Figure A–4). Once again the maximum possible number of interpretations is 6510 but this time the figure is reduced to only 18. These interpretations are a subset of the 21 made for the previous three images. The missing three are incorrect interpretations whose already large estimates of combined scale and stretch are now doubled and so no longer lie within the acceptable limits specified in the appropriate model part. The part selection process does not attempt to resolve the effects of scale and stretch and the combined scale and stretch parameters S_{gx}, S_{gy} and S_{gz} are estimated to be the correct values of 2.0, 2.0 and 2.0 for both the nose part and the face object.

The fifth image is much more interesting in that the synthetic face object has been deformed by stretching. As a further test, the stretching of the synthetic nose part differs from that of the synthetic face object. The nose is stretched by a factor of 1.1 along the Y–axis of the world coordinate frame which, since there is no rotation or translation between object and model, corresponds to the equivalent stretch along the y–axis of the model part coordinate frame. The face object has been stretched by a factor of 0.8, also along the Y–axis of the world coordinate frame again corresponding to the y–axis of the model coordinate frame.

This time the number of interpretations is pruned to 24. The extra three are incorrect interpretations previously just eliminated on the basis of their calculated scale and stretch but now the reduced distances between some of the object surfaces are sufficient for them to be retained.

The calculated values for the combined scale and stretch of the nose part $S_{gx_{nose}}$, $S_{gy_{nose}}$ and $S_{gz_{nose}}$ are not the expected values of 1.0, 1.1 and 1.0. This is because the three surfaces that make up the synthetic nose part form the symmetric, under–determined case discussed previously. By using the techniques developed to overcome this the coefficient of scale and stretch along the x–axis $S_{gx_{nose}}$ can be calculated exactly, but the two remaining coefficients, $S_{gy_{nose}}$ and $S_{gz_{nose}}$, have to be estimated yielding the "inaccurate" value of 1.078 for each.

The same problem occurs with the four selections of the synthetic face model.

For two of the selections the combined scale and stretch parameters $S_{gx_{face}}$, $S_{gy_{face}}$ and $S_{gz_{face}}$ are the expected values of 1.0, 0.8 and 1.0. However, the remaining two correct selections, the eyes and chin and the eyes and forehead surfaces, also form the symmetric, under–specified case and while $S_{gx_{face}}$ can be calculated exactly, $S_{gy_{face}}$ and $S_{gz_{face}}$ have to be approximated.

The last of the test images is the basic synthetic face object but with the chin surface showing a variation in position of $1.0mm$ in the direction of the $-Y$–axis of the world coordinate frame. Since there is no rotation of the object relative to the model, this corresponds to a displacement along the $-y$–axis of the model coordinate frame.

With the object so closely resembling that used in the first three images, it comes as no surprise that the maximum possible number of interpretations is pruned to the same 21 as before. Most interestingly however, are the deformation parameters calculated for the correct interpretations. The surfaces of the nose part show no variation in position and so the components of combined scale and stretch are the expected 1.0, 1.0 and 1.0. The four correct selections of the face object yield differing estimates of the scale and stretch of the face object. One combination of surfaces does not include the chin surface and so the calculated parameters are the same as those expected. Two selections that include the chin surface yield combined scale and stretch parameter that are close to nominal value of 1.0. The selection based on the eyes and chin surfaces, returns an exact, and correct, value for $S_{gx_{face}}$ but the values for $S_{gy_{face}}$ and $S_{gz_{face}}$ have to be estimated as close to 1.0.

The Class of Screwdrivers

The part selection process was then used with range data images obtained from real objects. The class of objects used here was a set of screwdrivers that exhibited varying degrees of scale and stretch. This class of object was interesting in that the structure of the object was relatively simple and this was reflected in the deformations shown by individual members of the class. The results of part selection for this class of object are given in Tables (6–4) and (6–5).

RESULTS FOR THE CLASS OF SCREWDRIVERS					
Image	Surfaces in image	Total no. interps	No. of object parts($>$ 3surf)	No. model selections	No. correct selections
Small Screwdriver	4	744	1	1	1
Medium Screwdriver	4	744	1	1	1
Large Screwdriver	4	744	1	1	1
Scaled Screwdriver	4	744	1	1	1
Multiple Screwdrivers	8	10416	2	2	2

Table 6–4: Summary of the Selections Made During Model Part Selection

The generic screwdriver model is based on the medium sized screwdriver. A full listing of this model may be found in Appendix B. The model is divided into two parts. The first is the handle and consists of three surfaces. The positions of these three surfaces are described, in the model part coordinate frame, in terms of x and z coordinates only, i.e. the centres of all three surfaces lie on a plane. The second part is the shaft and this consists of only one surface. The positions of both parts are defined in the model coordinate frame also in terms of only the x and z coordinates. This is not a complete representation of a screwdriver as only the surfaces on one side of the screwdriver are modelled. The surfaces on the other three sides of the screwdriver have been omitted to represent only those surfaces normally seen and to clarify the recognition process.

In all the images, each screwdriver is segmented into four surfaces corresponding to the four surfaces of the model. The shaft is represented (incorrectly) by a hyperbolic patch, the lower and middle handle areas are also represented by hyperbolic patches while the upper handle is represented by a positive elliptical surface.

The first image contains the "small" screwdriver (see Figure (A–8)). The surface

RESULTS FOR THE CLASS OF SCREWDRIVERS

Image	S_{gx}	S_{gx}	S_{gy}	S_{gy}	S_{gz}	S_{gz}
	Actual	Estim.	Actual	Estim.	Actual	Estim.
SMALL SCREWDRIVER Handle part	0.538	0.543	N/A	N/A	1.550	2.765
MEDIUM SCREWDRIVER Handle part	1.000	0.894	N/A	N/A	1.000	0.992
LARGE SCREWDRIVER Handle part	1.188	1.135	N/A	N/A	1.150	1.178
SCALED SCREWDRIVER Handle part	2.000	1.788	N/A	N/A	2.000	1.984
MULTIPLE SCREWDRIVERS Standard handle part	1.000	0.968	N/A	N/A	1.000	1.132
Large handle part	1.188	1.295	N/A	N/A	1.150	1.224

Table 6–5: The Deformation Parameters Estimated During Model PartSelection

data for this object is not well segmented, the central area of the handle in particular is not as expected. The four object surfaces in the image generate four different combinations of three surfaces. Each combination has 186 possible interpretations giving 744 possible interpretations in all. The application of constraints to the interpretation tree reduces this figure to just one, a correct selection of the screwdriver handle part. Because the interpretation tree for this object is relatively simple, the generation of the correct interpretation is shown in (Figure (6–6)). The shaft part of the screwdriver is *not* selected as the part selection process requires at least three surfaces of a part to be visible in the image before selection can take place. Were the surfaces for the other sides of the screwdriver to be modelled a wider interpretation tree with more interpretations would be expected.

The estimation of the combined scale and stretch parameters for the handle part, $S_{gx_{handle}}$ and $S_{gz_{handle}}$, is rather poor. There is no $S_{gy_{handle}}$ component for, as stated earlier, the model positions are defined in terms of only x and z. The poor estimates are due to the central region of the handle being segmented $5.0mm$ off the central axis of the screwdriver. Since the positions of the surfaces are modelled in terms of only x and z coordinates, a displacement in a direction equivalent to the model y axis is not expected. The increased distance between the objects surfaces results in greater than expected values for the combined scale and stretch.

The second image contains the "medium–sized" screwdriver (see Figure (A–7)). The generic screwdriver model was based on this screwdriver but is *not* an exact representation. As before the screwdriver object is segmented into four surfaces again giving 744 possible interpretations. Once again the application of constraints to the interpretation tree results in just one selection, a correct selection of the handle part. The correct interpretation is generated as shown in Figure (6–6).

The segmentation of this object is markedly better than that of the previous image. The estimates of the deformation parameters, $S_{gx_{handle}}$ and $S_{gz_{handle}}$, are close to that expected. With the resolution of the original range data along the axis of the screwdriver only $1.3mm$ per pixel and model distances as small as $9.0mm$, calculating the scale and stretch to be exactly that predicted, while still possible, is unlikely.

The third image contains the "large–sized" screwdriver (see Figure (A–9)). Of the

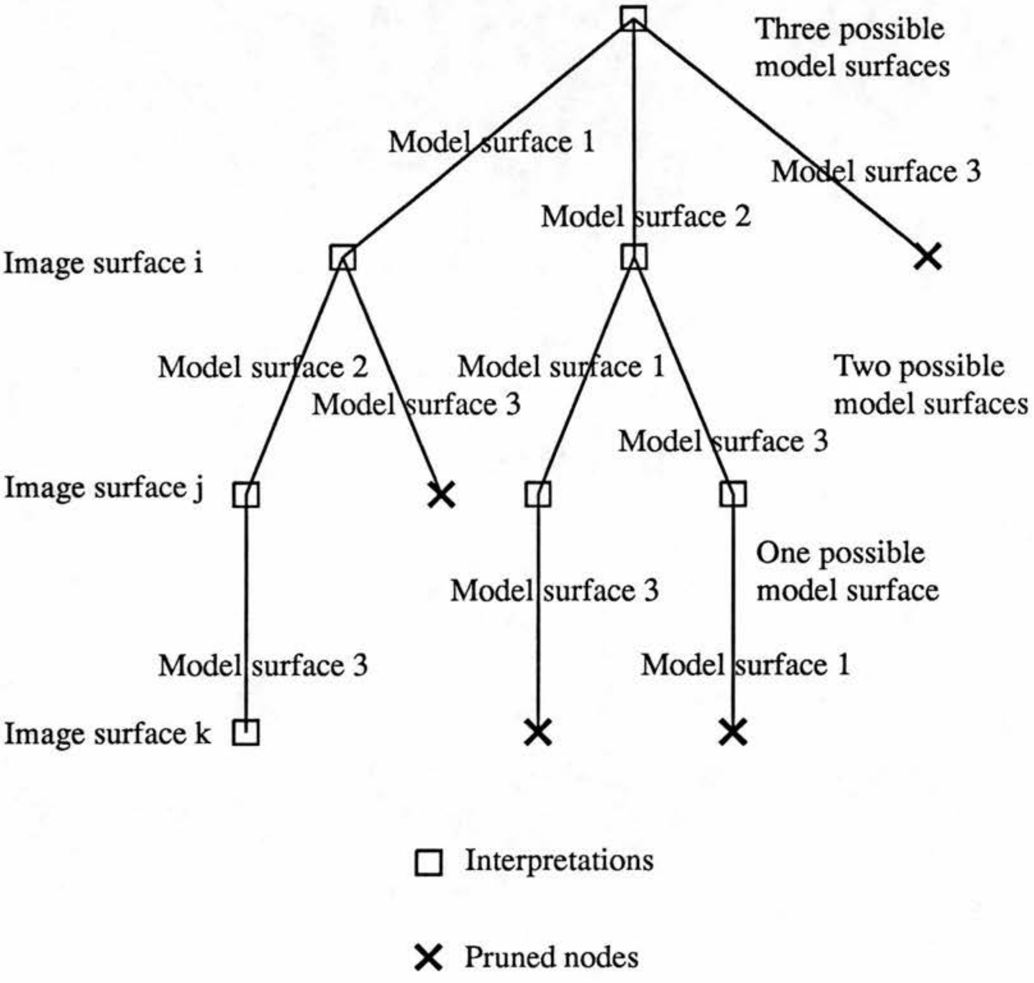


Figure 6–6: Generation of the Correct Interpretation of the Screwdriver Handle Part

744 possible interpretations only the correct interpretation of the three surfaces of the handle part remains after pruning of the search space. The segmentation of this image is also good and this is reflected by the accuracy with which the parameters of scale and stretch are calculated. The accuracy is surprisingly good when it is considered that the resolution of this image is down to $1.72mm$ per pixel along the axis of the screwdriver.

The fourth image contains the "medium–sized" screwdriver but the camera parameters have been changed so that the screwdriver appears scaled by a factor of two (see Figure (A–10)). Once again there are 744 possible interpretations and once again after pruning the interpretation tree only the one correct interpretation remains. Since this object is essentially the same object that is in Figure (A–7) but with a scaling factor introduced, the values estimated for the combined scale and stretch parameters reflect this.

The fifth image is interesting in that it contains two objects, the medium–sized screwdriver and the large–sized screwdriver (see Figure(A–11)). There are eight object surfaces in this image giving a possible 10416 interpretations. After pruning of the interpretation trees, this figure is reduced to two, correct interpretations of the two sets of three surfaces that make up each handle part. The segmentation of both objects is fair and the estimates of their parameters of deformation reasonable. The resolution of the image along the axes of the screwdrivers was once again $1.72mm$ per pixel.

The Class of Human Faces

This class of object consists of several parts, each of which exhibits varying degrees of deformation. Because the structure of each object is more complex than the previous, rather simple, class of objects there is more information available with which to determine the scale, stretch and variation of the object and each of its parts. The results of the part selection process for this class of object are given in Tables (6–6) and (6–7). Because of the large number of surfaces present in both the model and the object, it would take too much space to represent all the correct selections made, so a representative sample are given.

RESULTS FOR THE CLASS OF HUMAN FACES					
Image	Surfaces in image	Total no. interps	No. of object parts(> 3surf)	No. model selections	No. correct selections
Adult's Face 1	30	755160	3	872	113
Adult's Face 2	32	922560	3	537	12
Child's Face	27	544050	3	313	47

Table 6–6: Summary of the Selections Made During Model Part Selection

RESULTS FOR THE CLASS OF HUMAN FACES						
Image	S_{gx} True	S_{gx} Estim.	S_{gy} True	S_{gy} Estim.	S_{gz} True	S_{gz} Estim.
ADULT'S FACE 1						
Nose part	1.000	1.073	1.000	0.949	1.000	0.972
Eyes part	1.000	1.303	1.000	1.110	1.000	1.060
Face superpart	1.000	1.103	1.000	1.104	1.000	1.440
ADULT'S FACE 2						
Nose part	1.000	0.908	1.000	0.899	1.000	1.304
Eyes part	1.000	1.733	1.000	0.990	1.000	1.112
Face superpart	1.000	0.917	1.000	1.210	1.000	1.010
CHILD'S FACE						
Nose part	1.000	0.815	1.000	0.873	1.000	0.873
Eyes part	1.000	0.699	1.000	0.763	1.000	0.844
Face superpart	1.000	0.781	1.000	0.820	1.000	0.853

Table 6–7: The Deformation Parameters Estimated During Model PartSelection

The model representing the class of faces is based on the first human face image (see Figure (A–12)). The model is divided into eight surfaces and two parts – the eyes part and the nose part. The eyes part consists of a further eight surfaces while the nose part consists of five surfaces. Unlike the previous class of objects, the segmentation of each of the class examples shown here is different. However, most of the main features, particularly large areas like the forehead, chin and cheeks and areas of high curvature like the eye sockets and the tip of the nose are consistently segmented in all three images. It is combinations of distinctive features like these that the part selection process is intended to identify.

The first image is that of an adult human face (see Figure (A–12)). It is this face that forms a basis for the class model although the model is *not* an exact representation. The large number of surfaces in the image meant that a large number of combinations of object surfaces was unavoidably generated. The application of constraints to the interpretations found for each combination substantially reduced the number of interpretations. Since the model is based on this face, the shape and position of the model surfaces are similar to those of the object which explains the large number of correct selections made.

The complex shape and nature of the face meant that attempting to measure and predict the combined scale and stretch values of each part was difficult. The predicted values of unity are therefore nominal values indicating that no undue scaling or stretching is expected.

The second image is that of another adult (see Figure (A–13)). The segmentation of the range data in this face differs noticeably from the segmentation of the first. Several features identified in the first image, such as the eyeball, are absent from this image. Once again there are a large number of object surfaces visible and this leads to the large number of object surface combinations being generated. The much smaller number of selections made, compared to the first image, is due to several modelled features, such as the eyeballs and the chin, being absent from the image. As the image is that of an adult's face and no undue scale or stretch is expected, the nominal predicted values for the combined scale and stretch are again unity.

A problem arose with this image in that the three dimensional scale and stretch constraint eliminated several correct interpretations of the surfaces belonging to the eyes part. As represented in the model the surfaces that make up the eyes part are all separated by relatively small distances (a few millimetres) along the x–axis of the model part coordinate frame. The variation exhibited by the object surface patches of this face meant that small variations of only a few millimetres had a large enough effect on the estimates of combined scale and stretch that the estimated values fell outside the allowed limits. While this reduced the number of correct model selections, enough alternative interpretations remained to make a similar selection with different combinations of surfaces.

The final example of the class is a child's face (see Figure(A–14)). The segmentation of this image is similar to that of the first one with all the main features clearly segmented in both. There are slightly fewer object surfaces in this image and the resulting number of interpretations is slightly reduced. Although the image is segmented in a similar way to the first, fewer correct selections are made. This is mainly due to the forehead in this image being incorrectly classified as hyperbolic and the interpretations that include it being pruned by the shape constraint. Once again it is not possible to predict the combined scale and stretch parameters of each part but as the face appears "smaller" overall, the values obtained are expected to be less that the nominal value of unity representing an adult's face.

6.3 Model Superpart (or Model) Selection

The aim of the superpart (or model) selection process is to use correspondences between object and model parts or superparts to establish which model superparts, and eventually which model, are most likely to represent the object superparts, or the objects themselves, in the image. Although this process takes as input model parts and model superparts to select model superparts, larger model superparts or models, to avoid confusion the input here will be referred to as parts and the selection will be of superparts. The selection of larger superparts or models on the basis of superparts is achieved in exactly the same way.

6.3.1 Theory

As this process follows on from the model part and model superpart verification processes, the identities of the object parts which form the input to this process are already known. Since the identity of an object part is already known, selection of the corresponding superpart from the database is simple. However, if a part of an object superpart exists in the image, it is quite likely that further parts of the *same type of object superpart* may also be present in the image. These object parts could either be parts of different objects of the same type or different parts of the same object. Although each part could be used in isolation to select the appropriate superpart where more than one part of the same object exists, this leads to redundancy and the loss of evidence in identifying a correct selection. It would be better if those parts that might plausibly belong to the *same object superpart* could be identified and combined and the extra correspondences used to give a better interpretation.

Generating Combinations of Object Parts

As in the part selection process the approach of using combinations of object features is applicable here. However, since the identity of each object part is known, it is only necessary to generate combinations of the parts that could plausibly belong to the same object superpart. If there are P_i object parts in the image which are plausibly part of the same object superpart then the number of n–tuples that can be generated for a given object is given by:

$$number\ of\ combinations = \frac{P_i!}{(P_i - n)!n!} \tag{6.15}$$

If there are SP superparts for which at least one part appears in the image then the total number of n–tuples is:

$$total\ number\ of\ combinations = \sum_{i=1}^{SP} \frac{P_i!}{(P_i - n)!n!} \tag{6.16}$$

Since the number of superparts SP in an image is fixed, the number of combinations is dependent on the value of n. Ideally the value of n should be such that sufficient evidence can be obtained to identify a combination while keeping the number of combinations to an acceptable level. Because so much more is known about an individual object part than an object surface, combinations of a small number of object parts may still yield sufficient evidence for an accurate interpretation.

Interpreting Combinations of Object Parts

Each combination generated is interpreted by matching against permutations of model parts. Since each combination contains parts of the same object superpart they need only be matched with permutations of the m_i parts in the corresponding model superpart rather than all the parts in the database. The number of interpretations becomes:

$$total\ number\ of\ interpretations = \sum_{i=1}^{SP} \frac{P_i!}{(P_i - n)!n!} \frac{m_i!}{(m_i - n)!} \tag{6.17}$$

Provided that the object parts in a combination do belong to the same superpart and that the object superpart is represented by a model superpart that contains all corresponding parts, then the correct interpretation of that combination, and any other combinations which meet these criteria, will be generated.

Unlike the previous case of an object surface in isolation, the identity of a single object part is already known and can be used to select a superpart. The global scale of the object can be estimated, albeit poorly, from the combined scale and stretch parameters of the object part, the resolved stretch parameters of the model part providing at least an initial estimate for the stretch parameters of the superpart.

Two object parts, plausibly of the same object superpart, provide better information. As with a two surface correspondence, their separation provides an idea of the stretch parameters of the superpart independently of the stretch parameters of the constituent parts. Since the position and orientation of each part is known the scale and stretch parameters can be determined though their accuracy is dependent on the accuracy to which the orientations and positions of the parts were determined. Two independent sets of scale and stretch parameters enable a fair estimate of the global scale to be determined.

Three objects parts of the same object enable the scale and stretch parameters of the object superpart to be determined independently of the transformation of the component parts. This yields four sets of scale and stretch parameters and a reliable estimate of the global scale.

Reducing the Number of Interpretations

Because of the relatively small number of object parts that are present in an image and the fact that they need only be matched with a single known model superpart the number of possible interpretations will be relatively small. Nonetheless, constraints based on the limits of scale and stretch and the identities of the object parts can still be used to reduce the number of interpretations.

6.3.2 Implementation

The implementation of the model superpart selection process must address the issues descibed earlier. This requires the efficient generation and interpretation of combinations of object parts.

Generating Combinations of Object Parts

Regardless of the number of object parts present in an image, combinations are only generated for those that are part of the same type of superpart. Although the model superpart may contain many parts, not all the corresponding object parts may have been identified in the image whether due to having too few surfaces to originally have been selected, occlusion or misinterpretation. Rather than generate all possible combinations of three as before and be unable to identify any superpart with less than three parts present, all combinations of three, two and one parts are generated. Where there is a combination of three with a valid interpretation this leads to redundancy with three combinations of two and three combinations of one made making the same selection though with reduced evidence. The total number of combinations generated now becomes:

$$total\ number\ of\ combinations = \sum_{n=1}^{3} \sum_{i=1}^{SP} \frac{P_i!}{(P_i - n)!n!} \tag{6.18}$$

Interpreting a Combination Using an Interpretation Tree

To match each combination of object parts with each permutation of model parts an interpretation tree is again used. However, since the model superpart with which the object parts are to be matched is known, a tree is only generated for that superpart rather than each one in the model database. A further difference is that the depth of the interpretation tree varies with the number of parts in the combination, though the width of the tree, which is dependent on the number of parts in the model superpart, remains constant.

Estimating the Deformation Parameters in an Interpretation

The combined stretch and scale parameters for each object part have already been transformed into the superpart coordinate frame by the part verification process. Up until now, the effects of the global scale S_g and the components of part stretch S_x, S_y and S_z from all levels of the object have been combined as the parameters S_{gx}, S_{gy} and S_{gz}. Where parts have been identified, and especially where several parts are thought to belong to the same object, the scale of the object and the stretch of its constituent parts can be resolved, see Figures (6–7) and (6–8).

Where there is only one object part in a combination, the appropriate model superpart can be selected. The global scale S_g can be estimated as in Equation (8.21) but with only one set of combined scale and stretch parameters available the estimate is likely to be poor. With the global scale known, the components of stretch for the part can be resolved. In the absence of any further information, the stretch parameters of the part provide an estimate of the stretch parameters of the superpart.

Where there are two object parts in a combination, a fair estimate of the global scale can be estimated by Equation (8.21). Since the model verification process determined the coordinate frame transformation for each object part, the distances between the two parts in the object coordinate frame in terms of ΔX, ΔY and ΔZ can be related to the distances between the corresponding parts in the model coordinate frame in terms of Δx, Δy and Δz and the stretch parameters of the superpart determined directly, see Section 8.3.1. Unfortunately the accuracy with which the scale and stretch parameters

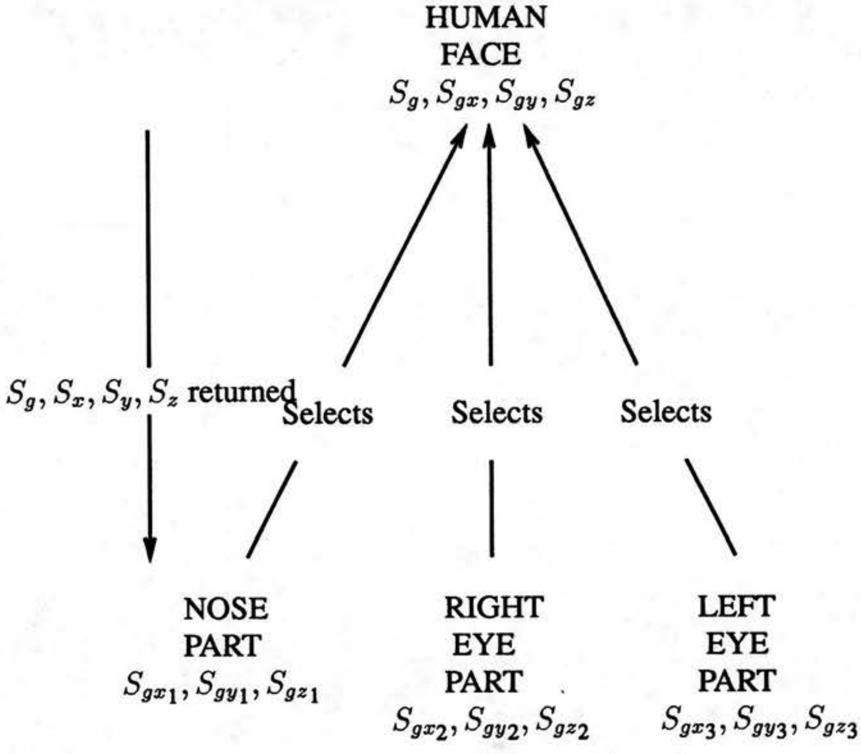

HUMAN
FACE
$S_g, S_{gx}, S_{gy}, S_{gz}$
S_g, S_x, S_y, S_z returned
Selects
Selects
Selects
NOSE
PART
$S_{gx_1}, S_{gy_1}, S_{gz_1}$
RIGHT
EYE
PART
$S_{gx_2}, S_{gy_2}, S_{gz_2}$
LEFT
EYE
PART
$S_{gx_3}, S_{gy_3}, S_{gz_3}$

Figure 6–7: Existence of Parts Select Superpart and Estimate Global Scale

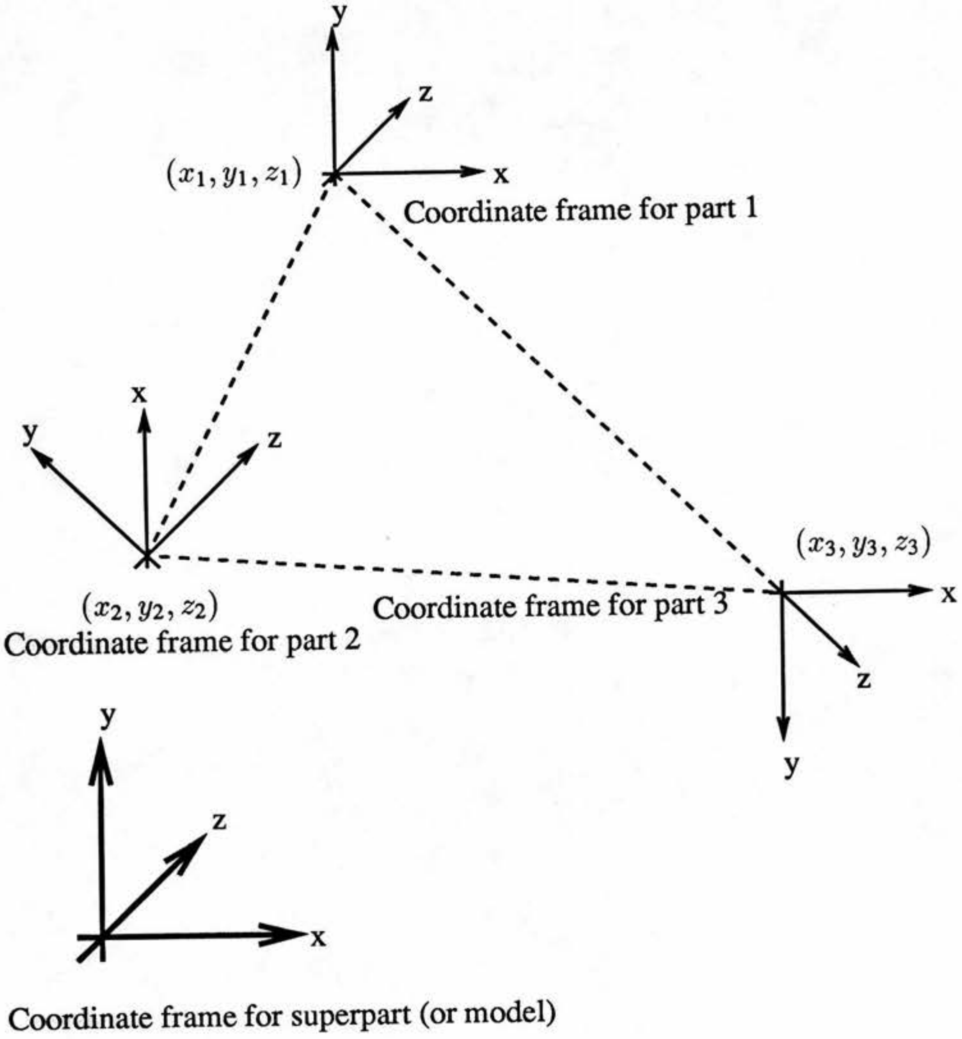

y
z
(x_1, y_1, z_1)
x
Coordinate frame for part 1
x
y
z
(x_3, y_3, z_3)
(x_2, y_2, z_2)
Coordinate frame for part 3
x
Coordinate frame for part 2
z
y
y
z
x
Coordinate frame for superpart (or model)

Figure 6–8: Estimating the Stretch Parameters of a Model Superpart

of the superpart are estimated is dependent upon the accuracy to which the coordinate frame transformations were determined.

A correspondence between three object and model parts enables the combined scale and stretch parameters of the superpart to be calculated, as described in Section 8.2.1, independently of the coordinate frame transformations of its constituent parts. The global scale can be estimated as in Equation (8.21), with four independent sets of scale and stretch parameters normally giving a good estimate of the global scale.

For each combination, if a valid interpretation exists then it will be generated. Because all combinations of three, two and one object parts are generated where a valid combination of three or two parts exists, there will also be combinations of two or one with the same interpretation though the values determined for S_g, S_{gx}, S_{gy} and S_{gz} may differ because of the individual variations shown by each part.

Ranking the Interpretations

Each interpretation is placed in the league table representing the appropriate level of the superpart in the corresponding model. Rather than rank the interpretations according to their degree of fit, they are ranked according to the *number of parts* in the combination. In this way the selections with the greater number of correspondences will be verified by the superpart verification process before those with fewer.

6.3.3 Reducing the Number of Interpretations

Because of the relatively small number of object parts that are present in an image and the fact that they need only be matched with a single known model superpart, the number of possible interpretations is usually small. Nonetheless, constraints can be applied to identify incorrect interpretations and reduce the size of the interpretation tree.

The Global Scale Constraint

Now that the global scale S_g has been resolved, it can be checked against the upper and lower limits of the global scale as defined in the model. The global scale is best estimated by using as many sets of stretch parameters as possible. This means that an incorrect interpretation will only be identified once all the correspondences have been generated and the interpretation tree already expanded to its maximum depth.

The Superpart Stretch Constraint

This is similar to the previous constraint in that each of the components of stretch for the object superpart must fall within the upper and lower limits imposed by the model superpart. The stretch parameters are not resolved until the global scale has been determined and so with this constraint too, incorrect interpretations can only be identified once all correspondences have been generated and the tree fully expanded.

The Part Identity Constraint

Since the identity of each object part is known it must be the same as that of the model part with which it is being matched. This is by far the most powerful constraint and can be applied at all levels of the interpretation tree. Taking the human face as an example, a combination including two object parts identified as noses could not be matched with the same model. However a combination containing two parts identified as eyes could plausibly be part of the same face and yield a valid interpretation.

Identifying a Valid Interpretation

Interpretations which meet the imposed constraints are regarded as valid interpretations. At this stage it is possible for a combination to have more than one interpretation though only one, or possibly none can be correct. A part may be present in several combinations with correct interpretations.

RESULTS FOR THE CLASS OF SYNTHETIC FACES				
Image	S_g	S_g	Number	Total No.
	Actual	Estim.	of Parts	of Features
STANDARD FACE				
Face Model	1.000	1.000	1	5
ROTATED FACE				
Face Model	1.000	1.000	1	5
TRANSLATED FACE				
Face Model	1.000	1.000	1	5
SCALED FACE				
Face Model	2.000	2.000	1	5
STRETCHED FACE				
Face Model	1.000	0.988	1	5
FACE WITH				
VARIATION				
Face Model	1.000	1.010	1	5

Table 6–8: The Global Scale Estimated During Superpart (Model) Selection

6.3.4 Results

As with the model part selection process, three sets of results are given in this section. The first is the set of synthetically generated images used to test the different functions of the superpart selection process independently. The second set illustrate superpart selection based on the minimum of evidence while the last set deals with superpart selection for a complex object.

The Class of Synthetic Faces

To test the various aspects of the superpart (model) selection process the synthetic images used before, were used here. The results of the superpart (model) selection process for these images are summarised in Tables (6–8) and (6–9).

RESULTS FOR THE CLASS OF SYNTHETIC FACES

Image	S_{gx}	S_{gx}	S_{gy}	S_{gy}	S_{gz}	S_{gz}
	True	Estim.	True	Estim.	True	Estim.
STANDARD FACE Face Model	1.000	1.000	1.000	1.000	1.000	1.000
ROTATED FACE Face Model	1.000	1.000	1.000	1.000	1.000	1.000
TRANSLATED FACE Face Model	1.000	1.000	1.000	1.000	1.000	1.000
SCALED FACE Face Model	2.000	2.000	2.000	2.000	2.000	2.000
STRETCHED FACE Face Model	1.000	1.000	0.800	0.800	1.000	1.000
FACE WITH VARIATION Face Model	1.000	1.000	1.000	1.010	1.000	1.052

Table 6–9: The Deformation Parameters Estimated During Superpart (Model) Selection

In the first synthetic face image (Figure (A–1)) both the face model and nose part were selected during model part selection. During the superpart (or in this case model) selection process the face model and nose part selected met the constraints imposed to be considered consistent with belonging to the same face object. As well as the four object surfaces identified as belonging to the face object, there was now a fifth feature identified, the nose part. With five correspondences the combined scale and stretch parameters of the face object were recalculated to take into account this new correspondence and agreed with the actual figures of 1.0, 1.0 and 1.0. With the combined scale and stretch of both the face object and nose part known this gave two independent sets of data with which to calculate the global scale. As can be seen, the value calculated agrees with the actual value of 1.0. The synthetic face model was selected for verification.

For the second (Figure (A–2)) and third (Figure (A–3)) synthetic face images the superpart selection process worked in exactly the same way as for the first image. The combined scale and stretch and global scale parameters were all calculated to be 1.0 and the synthetic face model is selected for verification.

The fourth image contains the scaled synthetic face (Figure (A–4)). The action of the superpart selection process was the same but this time each of the three parameters of the combined scale and stretch of the object was calculated to be 2.0, the correct value. The combined scale and stretch of the nose part was also 2.0 and using these two sets of data the global scale was also calculated to be 2.0.

The fifth image containing the stretched synthetic face (Figure (A–5)) shows a more interesting calculation of the global scale. Following the addition of an extra correspondence in the form of the nose part, the coefficients of combined scale and stretch of the face object $S_{gx_{face}}$, $S_{gy_{face}}$ and $S_{gz_{face}}$ are recalculated as 1.0, 0.8 and 1.0 respectively. On the other hand the combined scale and stretch parameters of the nose $S_{gx_{nose}}$, $S_{gy_{nose}}$ and $S_{gz_{nose}}$ are 1.0, 1.078 and 1.078 respectively. These two sets are averaged to give an estimate of the global scale of 0.988. The predicted global scale of 1.000 is a nominal value indicating that although the nose is enlarged and the face shrunk the overall scale of the face object is expected to be roughly the same – which is indeed the case.

RESULTS FOR THE CLASS OF SCREWDRIVERS

Image	S_g	S_g	Number	Total No.
	Actual	Estim.	of Parts	of Features
SMALL SCREWDRIVER				
Screwdriver Model	0.873	1.225	1	1
MEDIUM SCREWDRIVER				
Screwdriver Model	1.000	0.942	1	1
LARGE SCREWDRIVER				
Screwdriver Model	1.264	1.156	1	1
SCALED SCREWDRIVER				
Screwdriver Model	2.000	1.883	1	1
MULTIPLE SCREWDRIVERS				
Screwdriver Model	1.000	1.047	1	1
Screwdriver Model	1.264	1.259	1	1

Table 6–10: The Global Scale Estimated During Superpart (Model) Selection

The sixth image where the face exhibits individual variation (Figure (A–6)) has a similar solution. After the addition of the nose part to the face the combined scale and stretch parameters $S_{gx_{face}}$, $S_{gy_{face}}$ and $S_{gz_{face}}$ are recalculated as 1.000, 1.010 and 1.052 respectively. The parameters for the nose $S_{gx_{nose}}$, $S_{gy_{nose}}$ and $S_{gz_{nose}}$ are estimated to be 1.0, 1.0 and 1.0. Combining these two sets of data estimates the global scale to be 1.010. The predicted value of 1.000 is once again a nominal value reflecting the fact that the face has a scale of 1.000 excluding the effects of any variation which are expected, by definition, to be small.

The Class of Screwdrivers

During the part selection process each of the images containing a screwdriver had a single part, the handle part, identified. The exception to this was the image containing two screwdrivers in which two handle parts were identified. The identification of only a single part is the "worst case" for the superpart selection process. The combined scale and stretch parameters of the superpart (or in this case the model) cannot be

RESULTS FOR THE CLASS OF SCREWDRIVERS						
Image	S_{gx}	S_{gx}	S_{gy}	S_{gy}	S_{gz}	S_{gz}
	Actual	Estim.	Actual	Estim.	Actual	Estim.
SMALL SCREWDRIVER Handle part	0.538	0.543	N/A	N/A	1.550	2.765
MEDIUM SCREWDRIVER Handle part	1.000	0.894	N/A	N/A	1.000	0.992
LARGE SCREWDRIVER Handle part	1.188	1.135	N/A	N/A	1.150	1.178
SCALED SCREWDRIVER Handle part	2.000	1.788	N/A	N/A	2.000	1.984
MULTIPLE SCREWDRIVERS Standard handle part	1.000	0.968	N/A	N/A	1.000	1.132
Large handle part	1.188	1.295	N/A	N/A	1.150	1.224

Table 6–11: The Deformation Parameters Estimated During Superpart (Model) Selection

determined and the combined scale and stretch parameters of the part are used in lieu of anything better. With only a single set of data on which to base the estimate of the global scale the calculated value is likely to be inaccurate. The results of superpart (or in this case model) selection for this class of object are given in Tables (6–10) and (6–11).

For the image of the small screwdriver (see Figure(A–8)) the combined scale and stretch of the object is assumed to be the same as that of the part. This set of data also forms the basis for the estimate of the global scale. However, for the reasons discussed earlier the estimates for the combined scale and stretch of the handle part are inaccurate which results in the rather poor estimate of the global scale.

For the medium–sized screwdriver (see Figure(A–7)) the estimates of combined scale and stretch are more accurate resulting in a more reasonable estimate of the global scale.

The larger screwdriver (see Figure(A–9)) also gives a fair estimate of the global scale.

The scaled up screwdriver (see Figure(A–10)) gives, not surprisingly, a scaled up estimate of the global scale obtained from the image shown in Figure(A–7).

The image containing two screwdrivers (see Figure(A–11)) has two identified parts. However, since these are both handles they must come from two different objects. The estimates of global scale for these two screwdrivers are surprisingly good considering the limited amount of available data.

The Class of Human Faces

The model is based on the image containing the first adult face and as such it is no surprise that both the nose and eyes parts are identified as consistent with belonging to the same face object and are used in selecting the face model. These two parts when combined with the scale and stretch parameters of the face itself give a good estimate of the global scale.

RESULTS FOR THE CLASS OF HUMAN FACES				
Image	S_g	S_g	Number	Total No.
	Actual	Estim.	of Parts	of Features
ADULT'S FACE 1				
Face Model	1.000	1.055	2	10
ADULT'S FACE 2				
Face Model	1.000	0.902	1	4
CHILD'S FACE				
Face Model	1.000	0.766	2	10

Table 6–12: The Global Scale Estimated During Superpart (Model) Selection

RESULTS FOR THE CLASS OF HUMAN FACES						
Image	S_{gx}	S_{gx}	S_{gy}	S_{gy}	S_{gz}	S_{gz}
	True	Estim.	True	Estim.	True	Estim.
ADULT'S FACE 1						
Face model	1.000	1.083	1.000	1.101	1.000	1.330
ADULT'S FACE 2						
Face model	1.000	0.995	1.000	1.108	1.000	1.110
CHILD'S FACE						
Face model	1.000	0.832	1.000	0.815	1.000	0.793

Table 6–13: The Deformation Parameters Estimated During Model PartSelection

The second human face gives rather poor results. Because the eyeballs are not represented in the image, the verification process has been unable to verify the existence of the eyes part. As a result only the nose part and the few surfaces already identified as belonging to the face can be used in estimating the global scale and combined scale and stretch of the object. As a result the estimation of the parameters is rather poor.

The segmentation of the image containing the child's face is sufficiently similar to that of the first for both the nose and eyes parts to be selected and for these and the surfaces identified as belonging to the face to be determined as consistent with belonging to the same object. The resulting estimate of the global scale reflects the reduced scale of this face.

6.4 Comparison with Previous Work

The work described here is a combination of the local feature focus method [Bolles & Cain, 1982] and an interpretation tree search with cut–off [Grimson & Lozano-Pérez, 1987]. Focussing on individual object features is inappropriate since their appearance may have changed under object deformation. The use of a combination of object features overcomes this, provided that the object features are all from the same object. As this is not usually the case an interpretation tree search is used to find a consistent match (if one exists) between object and model.

Unlike traditional interpretation tree searches though, the tree is cut–off, only attempting to find a consistent match for three of the many features in the image. This approach is similar to that used on the TINA project [Pollard *et al*, 1987]. In that work the objects being matched did not change shape and so individual features were used to focus the recognition process. Not all image features were suitable for use as focus features and effort was expended searching for those that could be used to identify different areas of the image. Here, *all* object features are used in the combinations that act as focus features. In both pieces of work each focus feature caused an interpretation tree to be generated. In the TINA project the depth of the interpretation tree was limited in that interpretations were only sought for those features surrounding the focus feature in the image. The sets of correspondences formed were expected to contain incorrect matches but this was handled later. In the work described here the depth of the interpretation tree is limited to three. The interpretation trees produced are much greater in number, varying with the number of combinations of object features rather than linearly, but of less depth. With so few matches in an interpretation, if an object has at least three surfaces visible in the image at least one interpretation consisting entirely of correct matches will be generated. The difference becomes that of identifying the correct interpretations from amongst many verses identifying the correct clique of matches within an interpretation that is one of a relatively small number.

The use of constraints to reduce the search space represented by the interpretation tree has been covered extensively [Gaston & Lozano-Pérez, 1984], [Grimson & Lozano-Pérez, 198 [Drumheller, 1987], [Pollard *et al*, 1987], [Ikeuchi, 1987], [Lozano-Pérez *et al*, 1987], [Knapman, 1987], [Simsarian *et al*, 1990], and [Grimson, 1990b]. The use of such constraints for parameterised models is more rare the work described in [Murray, 1987b] and [Grimson, 1987] being the best examples. The shape, two dimensional scale and stretch, and triple scalar project constraints are similar to many used in previous work. The three dimensional scale and stretch constraint is unusual in that the the deformation parameters of the object are calculated and these used as the constraint rather than using constraints which are scale and stretch invariant. Although the computation required adds to the cost of the search it is worth it for the benefits gained later during the model verification process.

6.5 Conclusion

In this conclusion the processes of model part and model superpart selection are summarised. The efficiency with which interpretations are made and the accuracy of the estimates of deformation within those interpretations is discussed.

6.5.1 Summary

This chapter describes the processes of model part and model superpart selection. The aim of both process is to use object features to *efficiently* identify those objects in the image and determine their deformation parameters before selecting only the appropriate models for more rigorous matching.

Because of the deformation exhibited by different classes of objects, an individual object feature, in this case represented by a surface, yields little information as to its identity and that of the object to which it belongs. Instead of individual surfaces, combinations of three object surfaces are used. If each surface belongs to the same object then all three must be subject to the same object deformation which can then be estimated if the identity of the object part is assumed. An interpretation tree is used to match each combination of object surfaces in the image with each permutation of surfaces in the model part, superpart or model itself. The number of interpretations produced is large and constraints are developed to prune the interpretation tree of any invalid interpretations.

Superpart (or model) selection is accomplished in a similar manner. This time however the identity of each object part used to make a selection is known and the identity of the model superpart to be selected can be determined from this. Since there may be more than one object part from the same object superpart present in the image then before making a selection an interpretation tree is used to find combinations of object parts that are consistent with belonging to the same object superpart.

6.5.2 The Efficiency of Model Selection

For an image containing I object surfaces, possibly from different objects, and a model database containing p models (or model parts) each with m_i surfaces then the maximum number of interpretations of all possible combinations of three object surfaces given by the approach described previously is:

$$number\ of\ interpretations\ = \frac{I!}{(I-3)!3!}\sum_{i=1}^{p}\frac{m_i!}{(m_i-3)!} \tag{6.19}$$

The number of models, or model parts, and the number of surfaces within each is fixed so that the maximum number of interpretations is regulated by the number of combinations that, where I is much larger than 3, is approximated by I^3.

The standard interpretation tree search works most efficiently where all the features in an image belong to the same object. If this is the case then the maximum number of interpretations is given by:

$$number\ of\ interpretations\ = \sum_{i=1}^{p} m_i^I \tag{6.20}$$

if more than one image feature is allowed to match the same model feature.

If only one object feature is allowed to match each model feature then the maximum number of interpretations becomes:

$$number\ of\ interpretations\ = \sum_{i=1}^{p}\frac{m_i!}{(m_i-I)!} \tag{6.21}$$

The maximum number of interpretations produced by the method described above given by Equation (6.19) is *less* than the number given by Equation (6.20) where multiple assignments to model features is possible but *greater* than the figure given by Equation (6.21) where multiple assignments are also not allowed. A further disadvantage of the model selection process is that each interpretation generated only finds interpretations for three of the object surfaces. The traditional interpretation tree will have branches that represent interpretations of *all* the features in the image.

The standard interpretation tree approach is far less efficient where some of the features in the image do not belong to the same object, either spurious data or features of a completely different object. To account for image data not belonging to an object, when matching with a model, an extra branch is added to each node in the interpretation tree, to represent image data with no correspondence to the model. The maximum number of interpretations that can be generated now becomes:

$$number\ of\ interpretations\ = \sum_{i=1}^{p}(m_i + 1)^I \tag{6.22}$$

For an image containing spurious data, the maximum number of interpretations generated by the model selection process is still given by Equation (6.19). The maximum number of interpretations produced by the model selection process is now always less than the maximum number produced by the standard interpretation tree provided each object in the image has at least three features visible. The maximum number of interpretations produced by the model selection process is approximately a cubic function of the number of features in the image while the maximum figure for the original method is an exponential function. However, the interpretations produced by model selection are potentially less complete, only ever explaining three object features.

Because the performance of the model selection process is not degraded by the presence of other objects in the image, the corresponding models can (as here) be broken down into parts and each object part searched for individually. There is now an additional overhead in that object parts must be matched with model parts but as can be seen from the results the cost of this is small. The reduced number of features in a part rather than a whole object means that an interpretation of only three features presents less of a problem.

The number of interpretations discussed so far is the *maximum* number that can be generated. The use of local constraints enables the size of an interpretation tree to be reduced substantially from this maximum value. If the features in the image belong to the same object then the application of local constraints to a traditional interpretation tree reduces the number to a quadratic function of the number of model features. If the image features belong to more than one object then the search is far less efficient. With a carefully chosen heuristic and cut–off the number of interpretations generated becomes a polynomial function. In the model selection process the maximum number of interpretations is a cubic function *before* pruning of the search space. As shown by the results, this number can also be pruned dramatically by the use of the constraints developed.

6.5.3 The Determination of the Deformation Parameters

As demonstrated with the use of the synthetic image faces, the deformation of an object can be recovered regardless of the rotation and translation of the object parts in the image. In the, admittedly artificial, case of a perfectly uniform scaling operation with no stretch component, the effect of global scale is resolved correctly. The combined scale and stretch parameters of both parts and objects are accurately recovered and even in the symmetric case where an exact solution is not possible a reasonable estimate can be made. Where an object exhibits small individual variation then the combined scale and stretch parameters are calculated such that the variation is minimised. During superpart (model) selection of the synthetic face model reliable estimates of the global scale are determined from the two independent sets of combined scale and stretch parameters.

The objects present in real images are, unfortunately, far less obliging. The poor segmentation of the image containing the small screwdriver results in a poor estimate of the parameters of combined scale and stretch for the handle part. Since this is the only part to be identified this has a knock–on effect in the calculation of the global scale of the object. The combined scale and stretch of the medium screwdriver is expected to be unity. However, the positions of the segmented surfaces in the image are not quite as expected, due either to the surface segmentation or the rather coarse resolution of the image. Although the values of scale and stretch are other than those expected they are still close to the values predicted. The medium screwdriver appears again in the multiple screwdrivers image; while the the deformation parameters obtained are still not as expected they are much closer to those predicted. Similar problems occur with both images containing the large screwdriver. The calculated deformation parameters, while close to the predicted values, are not exactly the same.

Chapter 7

Model Verification

7.1 Introduction

For a recognition system to completely recognise and locate the objects in an image it is not enough to merely identify a few of the features belonging to each object and then to conclude that those objects are present in the image. This would give a poor estimate of each object's parameters and, where different objects share the same features, it could possibly give ambiguous identifications. Most importantly it would leave much of the image data unexplained. The hypothesised existence of a given object in the image must be proved or disproved by taking the corresponding model and verifying that all the features represented by the model either exist in the image or that their absence from it can be satisfactorily explained. Where a modelled feature is identified in the image then the extra evidence available from the correspondence means that the estimate of the object's parameters can be calculated more accurately.

This process of verification is essentially a combinatorial search problem. Given a number of unmatched model features and the object features represented in the image, all possible combinations of model to object correspondences should generated and tested to find the one that is correct. Unfortunately, even with models containing small numbers of features, the large number of combinations of possible model to object matches quickly leads to combinatorial explosion. Ideally, if the features in the image corresponding to the unmatched model features could be identified correctly in some way then the search space would be reduced.

Unfortunately, due to variations in the shape and positions of an object's features it is not possible to precisely predict and therefore identify an object feature with certainty. However, once some of the model features have been matched and an initial estimate of any model parameters made, it becomes obvious that many of the remaining features in the image are the wrong shape or in the wrong place to match any of those still unmatched in the model. These geometric constraints can be used to reduce the search space to an acceptable size. As more and more features are identified the model parameters can be determined with greater accuracy enabling plausible matches to be identified with greater confidence.

The models described here are hierarchical in structure. The lowest level of the model consists of parts which are defined purely in terms of surface primitives. At higher levels in the model, model parts and larger surfaces form superparts or the model itself. The model verification process is divided into two parts. The part verification process verifies the existence of an object part by seeking to establish model to object correspondences for each surface in the model part. Superpart (or model) verification aims to verify the existence of an object superpart (or the object itself) by establishing correspondences between the constituent parts and surfaces in the model with those in the image. This division is important for the way in which the object parameters are determined and the control flow of the two processes are radically different.

The process of model verification, as described here, is compared with previous research on the use of model based recognition systems, and the differences discussed.

7.2 Part Verification

The problem being addressed here is that of verifying that the object parts corresponding to the model parts selected during the part selection process are present in the image. As part of the verification process the positions of the object parts are determined and the deformation parameters updated.

7.2.1 Theory

The model part verification process is divided into three sections. Firstly, for the selected model part any remaining model to object correspondences have to be generated. Once the new correspondences have been obtained the position estimate of the model part must be updated. Finally, the problem of efficiency is addressed by attempting to eliminate incorrect correspondences.

7.2.2 Generating Model to Object Correspondences

Verification proceeds by attempting to find a corresponding surface in the image for each surface represented in the selected model part. Where an image contains I surfaces and a model part p contains m_p surfaces, if one or more model surfaces are allowed to correspond to the same object surface then the maximum number of possible verified matches is given by:

$$number\ of\ verified\, matches\ = I^m \tag{7.1}$$

For a given part only one of these sets will normally be correct.

Here the part selection process has already provided three object to model correspondences so the number of possible matches is reduced to:

$$number\ of\ verified\, matches\ = I^{(m-3)} \tag{7.2}$$

This set of matches must be generated for each selected model part for which verification is required. Provided that the initial selection was correct and that the object surfaces corresponding to the model surfaces represented are visible in the image, the correct set of matches will be generated.

7.2.3 Estimating the Position of an Object Part

Not all the surfaces represented by the model part may have a corresponding surface visible in the image. The matching surface might fall outside the field of view (the image) or be facing away from the viewer (camera). To predict the position and appearance of an object's surface it is first necessary to determine the deformation parameters and calculate the transformation between the model part and the object.

Because the models used here are allowed to deform, the determination of a rigid transformation between model part and object is not immediately possible. If, however, the model part can be deformed in the same manner as the object part, regardless of the rotation and translation between them, then the rigid transformation between the *deformed model part* and the object can be estimated.

The part selection process gives an initial estimate of the combined scale and stretch parameters of the selected model part and three object to model correspondences. If the model part is deformed by these parameters it should now show the same deformation as the object part. From the three correspondences an initial estimate of the coordinate frame transformation between the coordinate frame of the deformed model part and that of the camera, or world, coordinate frame may be calculated.

With the coordinate frame transformation between the model part and object known the shape, position and orientation of the object surfaces corresponding to any remaining unmatched model surfaces can be predicted.

Because of the individual variations normally exhibited by an object part in the shape, position and orientation of its surfaces, there is unlikely to be an *exact* estimate of the scale and stretch between model and part. The best that can be achieved is to calculate deformation parameters that are in some way the best approximation for the part as a whole. Without an exact deformation it is not possible to calculate an exact coordinate frame transformation. Once again the best approximation for the part as a whole must be estimated.

As new model to object matches are generated the extra evidence enables the current estimates of the deformation parameters and the coordinate frame transformation to be recalculated to more accurate values.

7.2.4 Reducing the Number of Matches

Part verification is required for almost all the parts selected during the part selection process. As with the part selection process the number of matches generated is large and the search for any correct matches is, at best, time consuming. As before the search space must be reduced to a practical size if the correct model to object correspondences are to be found in an acceptable time. By using the coordinate frame transformation between model and object the search for an unmatched surface can be restricted to those of a particular shape or orientation or those that lie in a particular area of the image. These constraints are "global" in that they are based on features common to the whole part and determined using all the correspondences found so far. A further difference from the local constraints described in the previous chapter is that these are coordinate frame dependent.

The variations present in the shape, position and orientation of individual object parts means that geometric constraints where the parameters of an expected object feature are *known* to lie within a small error volume are of limited use here. The appearance and position in the image of an unmatched model surface can be only roughly predicted. However, whether due to variation in the object surface being looked for or variation in the surface data originally used to make the prediction, it can be the case that no single surface may meet the constraints of the predicted shape and position. In this case several possible matches may have to be considered further. If larger numbers of model to object correspondences are used then the effects of individual variations are reduced and the predictions become more accurate. In the early stages of part verification, where there are few model to object matches, the accuracy with which unmatched surfaces can be predicted is usually poor and a greater number of possible matches will be accepted. In the latter stages of the verification process, as more correspondences are achieved, the estimates of stretch, rotation and translation can be recalculated with greater accuracy. With more accurate predictions, the constraints that a possible match must achieve can be tightened enabling the search space to be reduced still further.

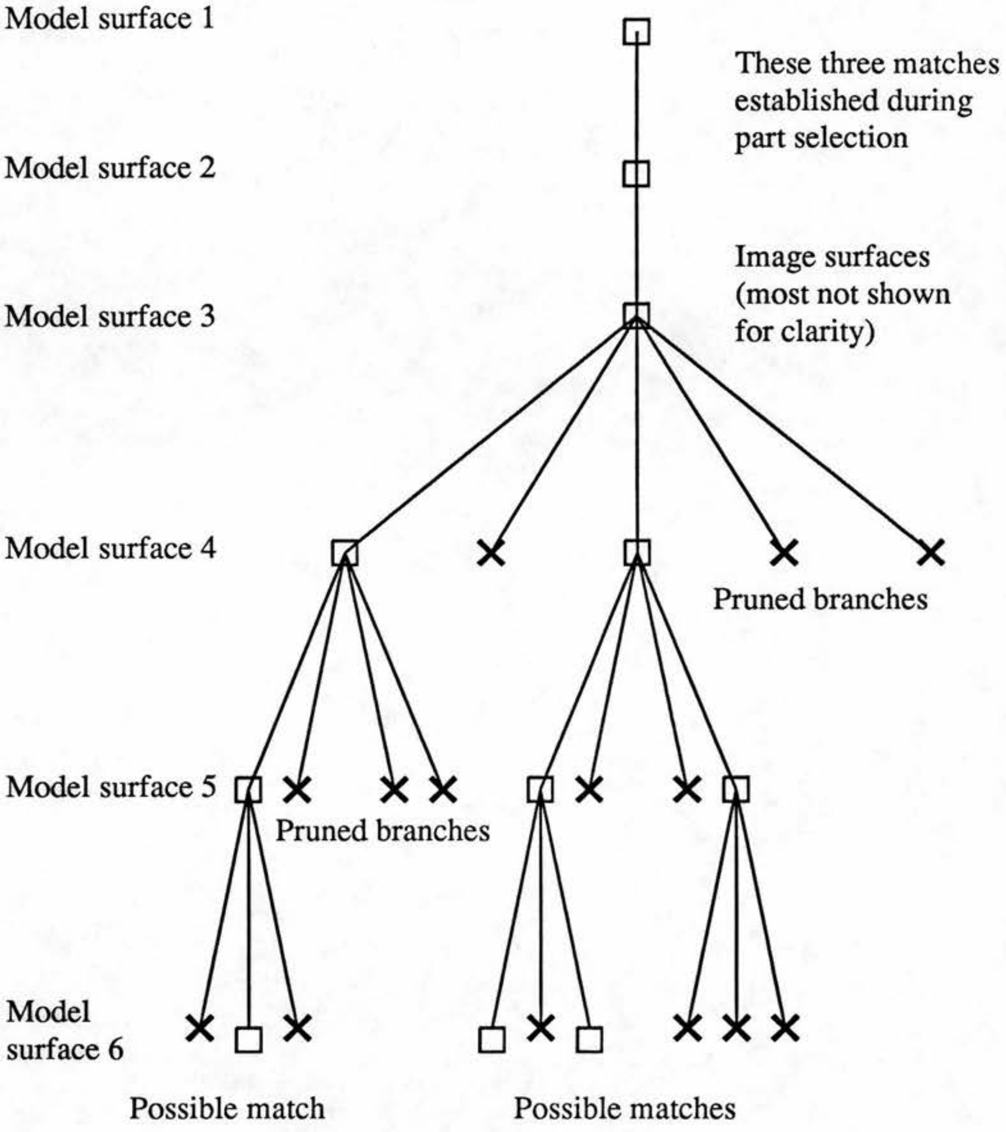


Figure 7–1: The Search Tree Used for Part Verification

7.2.5 Implementation

To represent the possible matches between the unmatched surfaces represented in the model and the object surfaces in the image a search tree is once again used. This tree differs from the interpretation tree used in the previous chapter in that the width of the tree represents the number of surfaces in the image and the depth of the tree represents the number of surfaces in the model part, see Figure (7–1). A further difference is that the depth of the tree is not fixed but will vary with the number of surfaces contained in the model part. Since three correspondences have already been supplied by the part selection process, the depth of the tree will be the total number of surfaces less three. Where there are only three surfaces represented in the model, only the root node of the tree, where the coordinate frame transformation is calculated, is required. One search tree is generated for each part selection for which verification is required.

Starting the Model Part Verification Process

The model part verification process starts by taking the current unverified leader in the league table containing parts produced by the model part selection process. This provides three object to model surface correspondences, the identity of the object part to which the object surfaces belong and a first estimate of the combined scale and stretch parameters S_{gx}, S_{gy} and S_{gz}. As described previously, there are certain cases where one of the stretch parameters may not be known.

The model part is deformed and rigid coordinate frame transformation between the deformed model part and object part established. If there are more than three surfaces in the selected model part then these must either be identified in the image or their absence from it explained. Each missing surface is searched for in turn. Where a new correspondence is achieved this enables a more accurate estimate of the scale and stretch parameters and the coordinate frame transformation to be determined before seeking correspondences for any remaining model surfaces.

Estimating the Deformation of the Object Part

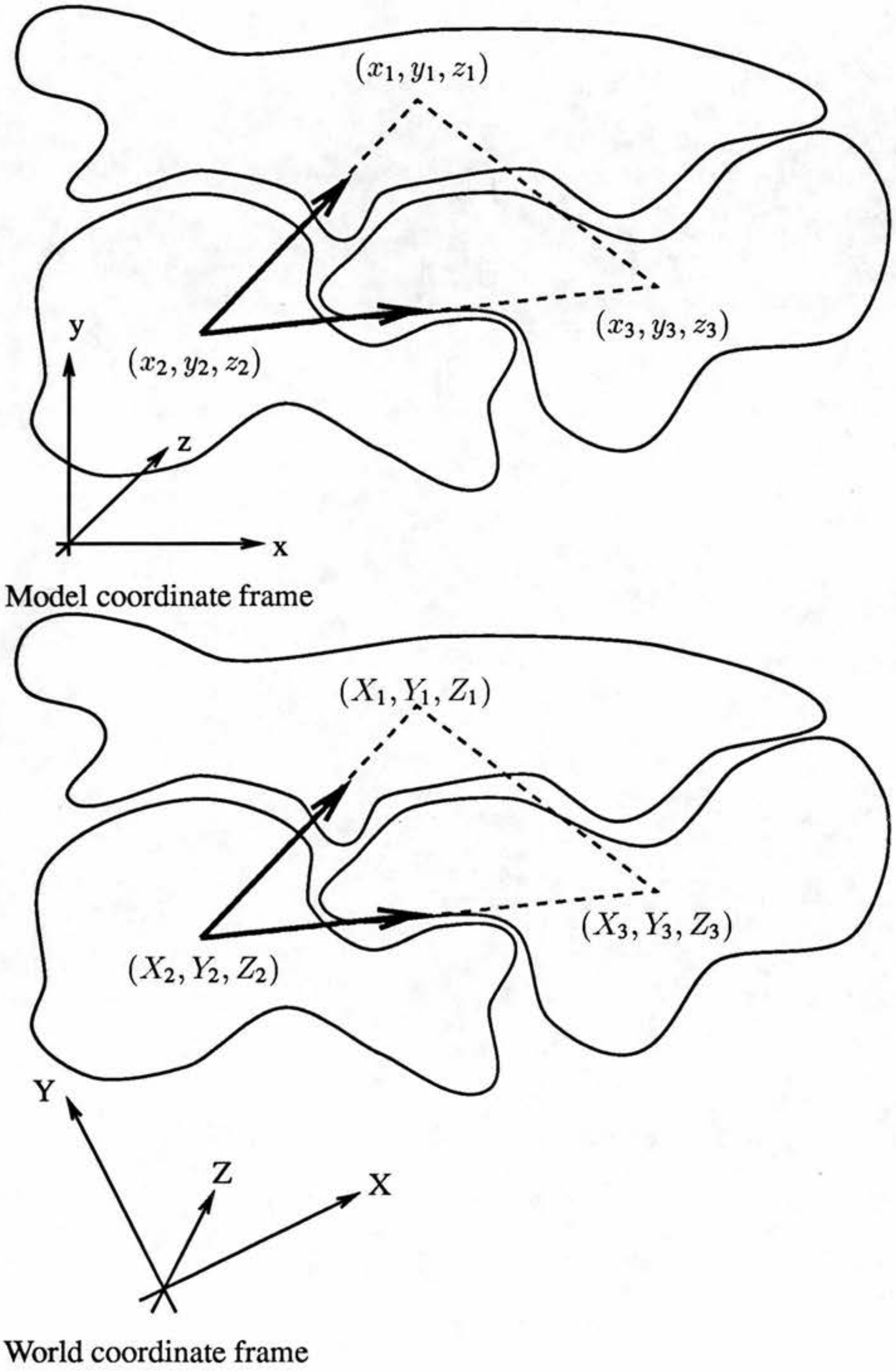
The combined scale and stretch factors for the initial three model to object correspondences are provided by the part selection process. However, when a new correspondence is established, the combined scale and stretch parameters of the model part must be recalculated to minimise the effects of individual variation for each surface in the part. Each set of model part surface to object surface correspondences gives a set of equations in the form of equation (6.10). With four or more correspondences, the set of simultaneous equations produced becomes an over–specified problem. The best estimates of the updated scale and stretch coefficients S'_{gx}, S'_{gy} and S'_{gz} are given by solving the entire set of equations using a least–squares method as detailed in Section 8.4.1.

Deforming the Model Part

Although the estimated combined scale and stretch factors deform all surfaces in the model, only the next model surface for which the corresponding surface in the image is to be identified need be subjected to the deformation transformation (4.4). If a correspondence for the model surface is achieved then the coefficients of scale and stretch are updated as described previously by the least–squares method. Only now is it required that all the model surfaces corresponding to surfaces in the image be deformed by the latest estimates of the scale and stretch. The updated positions of these surfaces in the deformed model part coordinate frame are denoted $(x'_{di}, y'_{di}, z'_{di})$.

Estimating the Rotation of the Deformed Model Part

The three correspondences between the deformed model part and the object initially supplied by the part selection process provide two sets of two vectors (see Figure (7–2)). These are the minimum required to establish the rotation between the model part coordinate frame and that of the world $R_{pw}(r, s, t)$. If a further model to object correspondence is established then the rotation will need to be recalculated to take account of it. As occurred in the recalculation of the scale and stretch coefficients, with four or more correspondences, the problem becomes over–determined. Due to the effects of individual variation in the positions of the object's surfaces, an exact rotation that rotates each deformed model surface to the same orientation as the object surface is unlikely. The best estimate of the updated part rotation $R_{pw}(r', s', t')$ is obtained by matching each set of vectors using a least squares fit as detailed in Section 8.4.1. For efficiency the inverse rotation from the world coordinate frame to model part $R_{wp}(R', S', T')$ is determined at the same time and stored for future use.



Model coordinate frame

World coordinate frame

Figure 7–2: Two Vectors Determine Rotation Between Deformed Model and Object

Rotating the Deformed Model Part

Having calculated the rotation between the coordinate frame of the deformed model part and that of the world, initially only the next model surface for which a correspondence is required need be rotated. If a corresponding object surface is identified then the rotation must be recalculated using the least–squares method as described earlier. All model surfaces that represent object surfaces in the image are now subjected to this updated rotation estimate. The updated positions of the surfaces in the deformed and rotated model part are denoted $(x'_{dri}, y'_{dri}, z'_{dri})$.

Estimating the Translation of the Deformed Model Part

Having deformed and rotated the model part it only remains to find the translation between the model part and world coordinate frame t_{pw_x}, t_{pw_y} and t_{pw_z}. Since the three correspondences initially supplied by the part selection process assume individual variation in the object surface position to be negligible, there will be a translation that exactly maps the position of each of the three deformed and rotated model surfaces $(x'_{dri}, y'_{dri}, z'_{dri})$ to the positions of the corresponding object surfaces (X_i, Y_i, Z_i). However, as further correspondences are established then the problem becomes over–determined and individual variations in the object surface positions mean that the determination of an exact translation is no longer possible. The best estimate of the updated translation of the deformed and rotated part t'_{pw_x}, t'_{pw_y} and t'_{pw_z} is obtained as detailed in Section 8.4.1. To save time, the translation between the world coordinate frame and that of the model part t'_{wp_X}, t'_{wp_Y} and t'_{wp_Z} is calculated at the same time.

Translating the Deformed Model Part

Since a correspondence is only required for one model surface at a time, only this surface need be translated initially. Once a correspondence has been achieved then, as described in Section 8.4.1, the translation must be recalculated. All surfaces in the deformed and rotated model part that are matched with object surfaces are translated by this updated estimate. The translated model points are represented by

$(x_{drti}, y_{drti}, z_{drti})$. These will be *close* to the positions of the corresponding object surfaces (X_i, Y_i, Z_i) but, except in the case of only three corresponding surfaces or an object whose parts show no variation (normally a synthetically generated image), not exactly the same.

Estimating the Individual Variation

As stated, the positions of the deformed, rotated and translated model surfaces $(x'_{drti}, y'_{drti}, z'_{drti})$ are *not* the same as the positions of the corresponding object surfaces (X_i, Y_i, Z_i), but the positions predicted if the object is subject only to scaling, stretching, rotation and translation. Any difference is due to either the position of a surface being determined incorrectly, for example by poor segmentation, or by the individual variation in object surface position V_{xi}, V_{y_i} and V_{zi}. The variation of a surface is individual to that surface and so is merely the difference between that predicted by the model and that actually shown by the object. Unlike the other forms of deformation, individual variations in the position of a surface are not related to the variation shown in its other properties such as its curvature.

Application of the Individual Variation

Since the variation shown by a surface is individual to that surface, then it is of little use in predicting the properties of a different surface and so is not usually used in the search process. The one time when it is used is where a model contains two (or more) identical surfaces arranged symmetrically in the model part coordinate frame. Since the surfaces are identical in the model it is assumed that they will be identical in the image. The individual variations between one of model surfaces and the corresponding object surface should be present in the identical model surface. When predicting the shape and position of these surfaces then the individual variations *are used* to modify the parameters of each surface accordingly. This is the *only* time that these parameters are used during the recognition process.

Surface Prediction

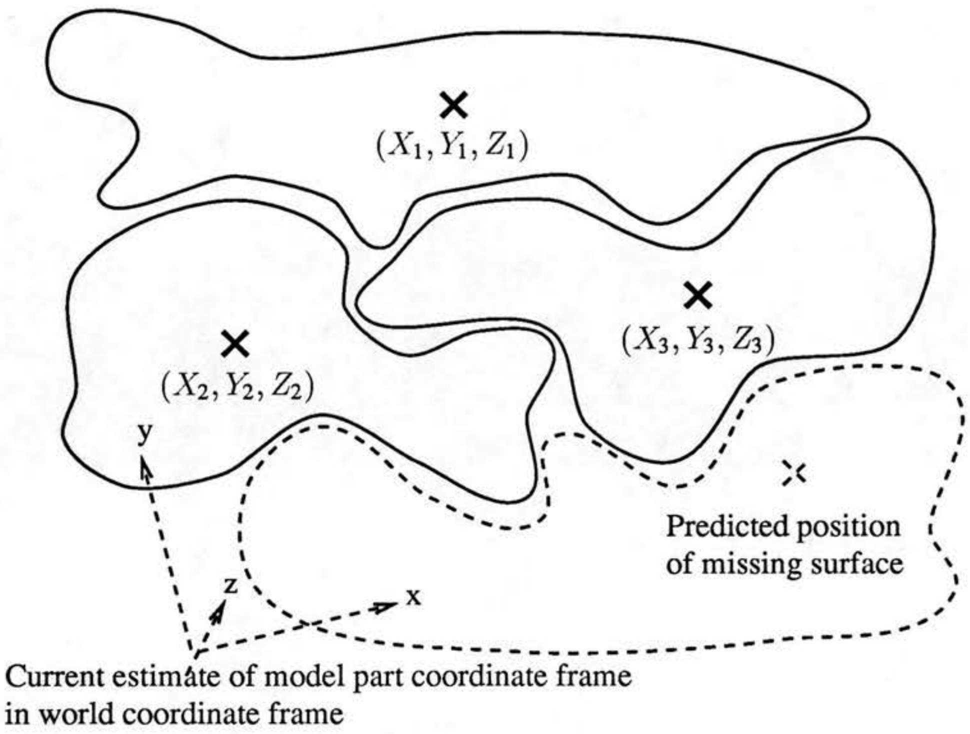

(X_1, Y_1, Z_1)
(X_3, Y_3, Z_3)
(X_2, Y_2, Z_2)
y
z
x
Predicted position of missing surface
Current estimate of model part coordinate frame in world coordinate frame

Figure 7–3: Prediction of a Fourth Surface.

Given that the combined scale and stretch of the model part and the coordinate frame transformation between the deformed model and the world coordinate frames is known, it is possible to predict the appearance in the image of the object surfaces corresponding to those model surfaces currently unmatched. The next model surface in the model part is deformed, rotated and translated by the latest estimates of the combined scale and stretch and the coordinate frame transformation to predict the shape, position and orientation in the image of the corresponding object surface (see Figure 7–3).

The transformed model surface is assumed to be visible in the image if its predicted position falls within the limits of the image and some component of its predicted surface normal vector points towards the viewer (camera). If the predicted surface fails to meet either of these criteria then it is tagged as absent from the image. In this case the image is *not* searched for a corresponding surface and the next unmatched model surface (if any) is taken. If the predicted surface should be visible then the search tree is expanded and the image searched for possible correspondences (see Figure 7–4).

Occasionally the part selection process is only able to establish two of the combined

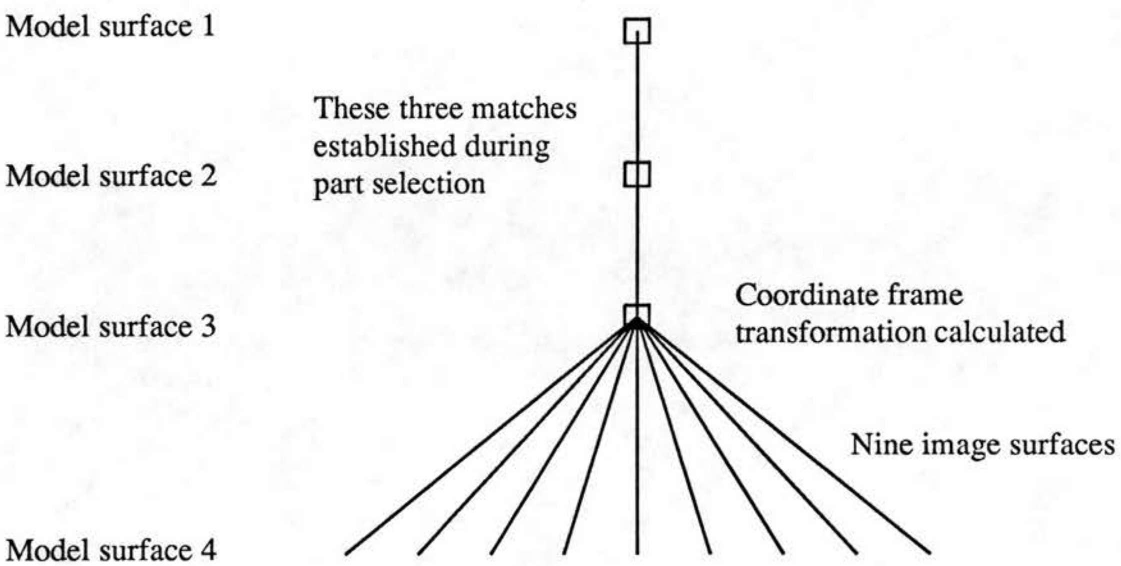

Model surface 1
These three matches established during part selection
Model surface 2
Model surface 3
Coordinate frame transformation calculated
Nine image surfaces
Model surface 4

Figure 7–4: Expanding the Search Tree Searching for a Fourth Surface

scale and stretch factors. In this case the positions of any remaining surfaces can only be predicted as lying along a line in the world coordinate frame.

Identification of a Correct Correspondence

The degree to which a surface in the deformed, rotated and translated model part fits an object surface in the image is determined by how closely the position of the object surface matches that predicted by the model. The degree of fit between the two is given by:

$$\begin{aligned} degree\ of\ fit \; = \alpha \; & ((X_i - x_{drti})^2 + \\ & (Y_i - y_{drti})^2 + \\ & (Z_i - z_{drti})^2) \end{aligned} \qquad (7.3)$$

where α is the position weighting coefficient discussed in chapter 3 and (X_i, Y_i, Z_i) and $(x_{drti}, x_{drti}, x_{drti})$ are the positions of the object and predicted model surfaces respectively.

A correct correspondence is assumed if the degree of fit falls below a threshold T. (The threshold used here is $\frac{w}{4l}$ where w is the distance represented by the width of the image which is an orthographic projection and l is the level of the node in the search tree). Where one of the components of scale and stretch is unknown then the appropriate term is removed from the degree of fit function.

Expanding the Search Tree

If one or more object surfaces are sufficiently close to the predicted position then the correspondence is assumed to be correct and the deformation and coordinate frame transformation recalculated accordingly. A further constraint is now applied in that the coefficients of combined scale and stretch should not fall outside the limits imposed in the model part. Those nodes in the search tree which represent a correct correspondence are tagged to be expanded further. All other nodes at this level of the tree are discarded, or pruned, from the search, see Figure (7–5).

If none of the surfaces in the image are sufficiently close to the position predicted from the model or fail to meet the scale and stretch constraints then the search terminates, the model part selection is assumed to have been incorrect and is deleted from further processing.

Ideally, if the object corresponding to the selected part does not exist in the image the geometric constraints should eventually eliminate all the remaining surfaces in the image as plausible candidates so pruning the tree so that no complete branch remains. At this point both the search tree and the original selection are discarded. If the object does exist then in the absence of symmetric cases there should ideally be only one branch in the tree remaining unpruned.

At the new node in the search tree the estimated coordinate frame of the object part is recalculated to take account of the new correspondence, see Figure (7–6).

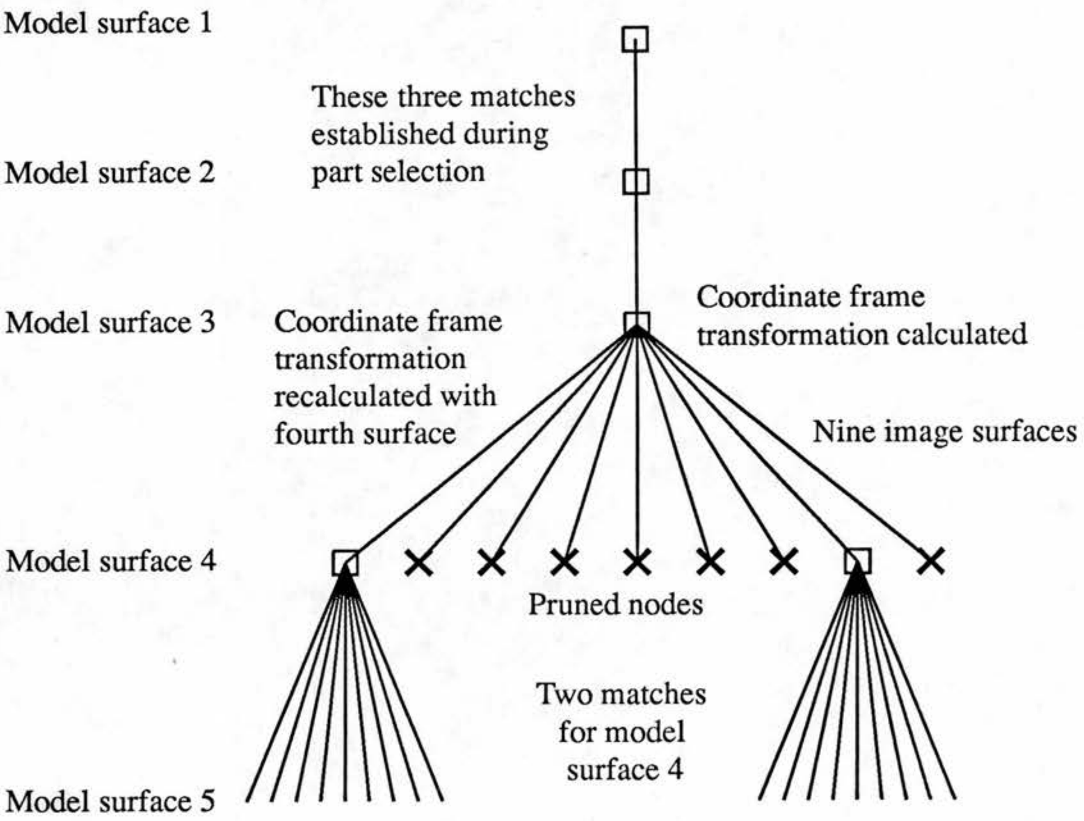

Model surface 1
These three matches established during part selection
Model surface 2
Model surface 3
Coordinate frame transformation recalculated with fourth surface
Coordinate frame transformation calculated
Nine image surfaces
Model surface 4
Pruned nodes
Two matches for model surface 4
Model surface 5

Figure 7–5: Expanding the Pruned Search Tree

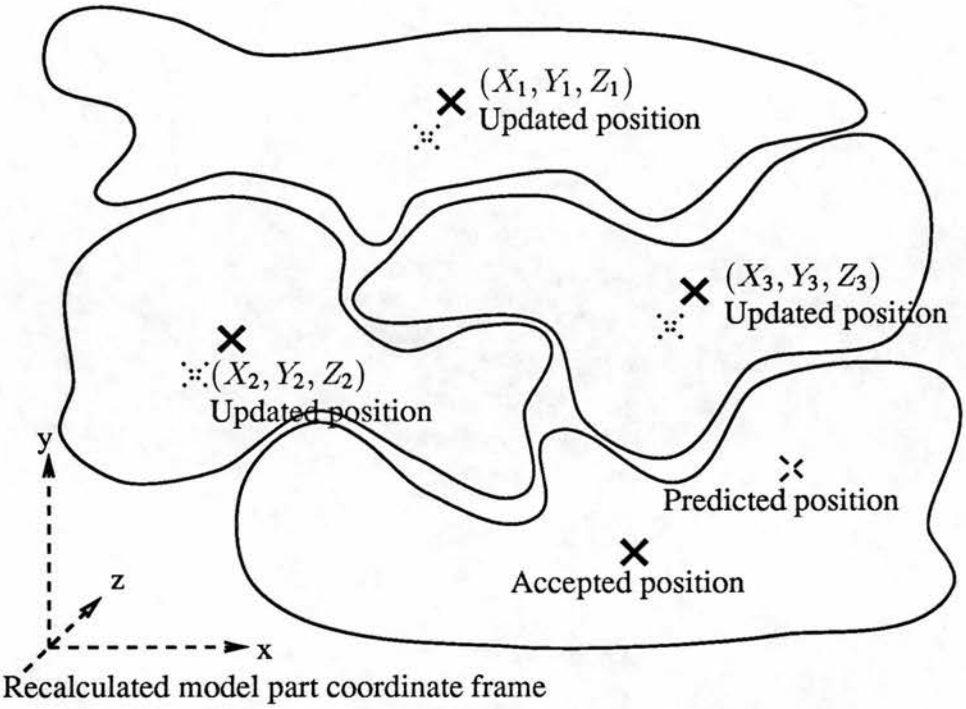


Figure 7–6: Recalculation of the Coordinate Frame.

Completion of the Part Verification Process

Verification of the selected model part is achieved when *all* the surfaces in the model part have corresponding surfaces in the image or their absence from the image has been explained. Ideally, the lowest level of the search tree will have only one expanded node yielding only one set of model to object correspondences. It is possible however that several nodes will remain expanded yielding several plausible sets of correspondences. Where this is the case the degree of fit of each set is evaluated and all but the best deleted.

The model contains the geometric relationship between the coordinate frame of the verified model part and the coordinate frame of the superpart (or model) to which it belongs. The final estimate of the combined scale and stretch factors are transformed into the superpart coordinate frame for use by the superpart selection process.

The initial model part selection was achieved using correspondences of only three of the model surfaces. Where, as is usually the case, a model part consists of more surfaces than this, then the same model part could have been selected several times by different combinations of the object surfaces matching with different permutations of its surfaces. The more surfaces that a model part has, the more likely this is to happen. Once the existence in the image of a model part has been verified a search is made of the appropriate league table for any lower ranked and as yet unverified selections of the same model part. Those selections that have *all three* correspondences in common with the correspondences of the part just verified are assumed to be redundant selections and are deleted from the table. Those selections where only one or two of the correspondences are the same are *not* deleted.

Once all the model part selections in the list have either been verified or deleted from further processing they are placed in groups according to the type of superpart of which they are a part. These groupings are then passed to the superpart selection process which seeks to select model superparts or models from configurations of these verified parts.

7.2.6 Results

Three sets of results are presented here. The first set of results were obtained from a set of synthetically generated images designed to test the various aspects of the model part verification process. The second set of results are from a class of real, but extremely simple, objects, while the last set of results are from a set of complex objects.

The Class of Synthetic Faces

To test the various aspects of the part verification process the six synthetically generated images introduced previously were used. The results of the part verification process for these images are summarised in Tables (7–1), (7–2), (7–3), (7–4) and (7–5).

The first of these images contains the basic synthetic face. There is no rotation or scaling of the object and the only translation is along the Z–axis. Of the 21 selections originally made eight are successfully verified by the verification process. A further three were found to be subsets of a previously verified selection and were deleted without verification. Of the eight that were verified these include the two correct

RESULTS FOR THE CLASS OF SYNTHETIC FACES					
Image	Number of Selections	Redundant Selections	Selections Verified	Incorrect Verifications	Correct Verifications
Standard Face	21	3	8	6	2
Rotated Face	21	3	8	6	2
Translated Face	21	3	8	6	2
Scaled Face	18	3	4	2	2
Stretched Face	24	3	7	5	2
Face with variation	21	3	11	9	2

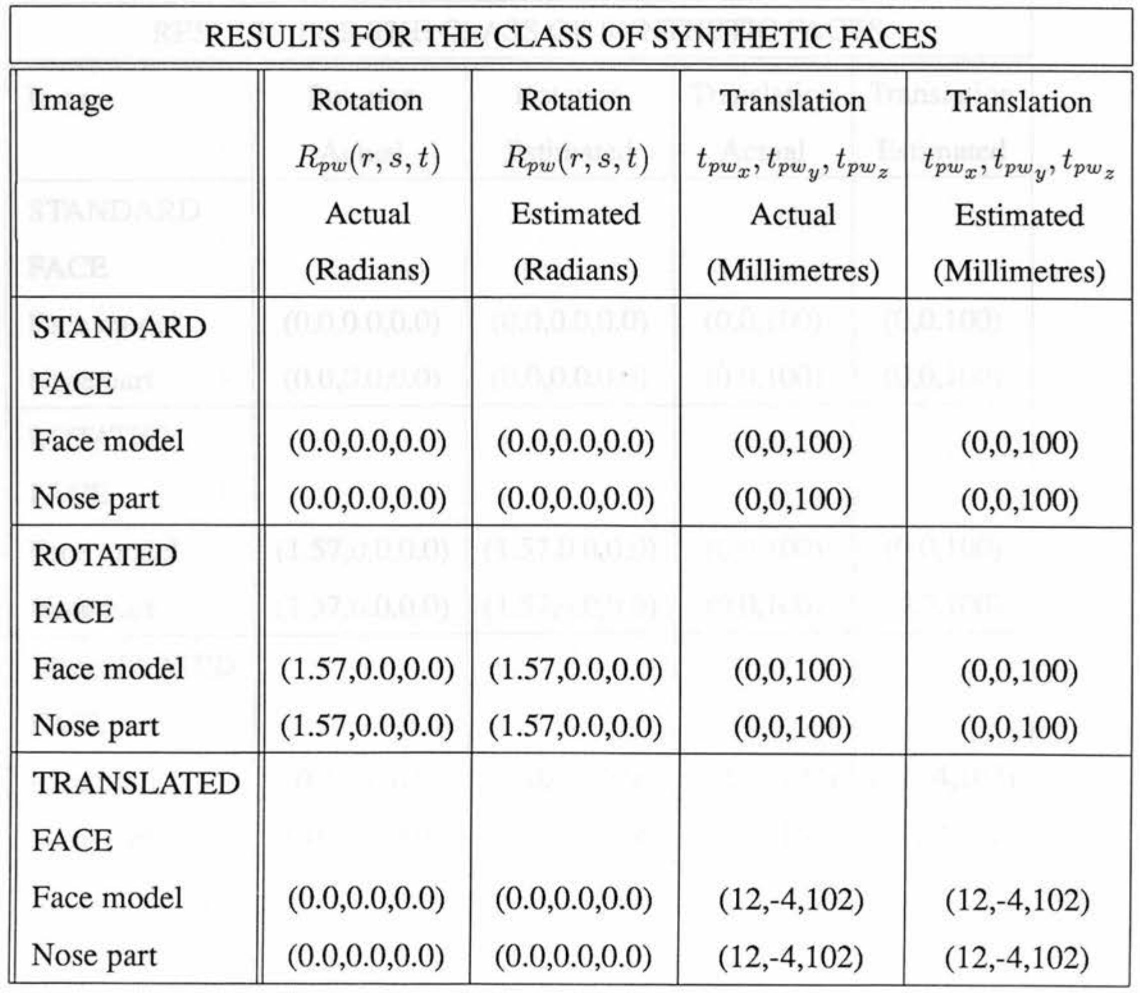
Table 7–1: Summary of the Verifications Made During Model Part Verification

RESULTS FOR THE CLASS OF SYNTHETIC FACES						
Image	S_{gx} True	S_{gx} Estim.	S_{gy} True	S_{gy} Estim.	S_{gz} True	S_{gz} Estim.
STANDARD FACE						
Nose part	1.000	1.000	1.000	1.000	1.000	1.000
Face model	1.000	1.000	1.000	1.000	1.000	1.000
ROTATED FACE						
Nose part	1.000	1.000	1.000	1.000	1.000	1.000
Face model	1.000	1.000	1.000	1.000	1.000	1.000
TRANSLATED FACE						
Nose part	1.000	1.000	1.000	1.000	1.000	1.000
Face model	1.000	1.000	1.000	1.000	1.000	1.000

Table 7–2: The Deformation Parameters Estimated During Model Part Verification

RESULTS FOR THE CLASS OF SYNTHETIC FACES						
Image	S_{gx} True	S_{gx} Estim.	S_{gy} True	S_{gy} Estim.	S_{gz} True	S_{gz} Estim.
SCALED FACE						
Nose part	2.000	2.000	2.000	2.000	2.000	2.000
Face model	2.000	2.000	2.000	2.000	2.000	2.000
STRETCHED FACE						
Nose part	1.000	1.000	1.100	1.078	1.000	1.078
Face model	1.000	1.000	0.800	0.800	1.000	1.000
FACE WITH VARIATION						
Nose part	1.000	1.000	1.000	1.000	1.000	1.000
Face model	1.000	1.000	1.000	1.010	1.000	1.050

Table 7–3: The Deformation Parameters Estimated During Model Part Verification



RESULTS FOR THE CLASS OF SYNTHETIC FACES				
Image	Rotation $R_{pw}(r,s,t)$ Actual (Radians)	Rotation $R_{pw}(r,s,t)$ Estimated (Radians)	Translation $t_{pw_x}, t_{pw_y}, t_{pw_z}$ Actual (Millimetres)	Translation $t_{pw_x}, t_{pw_y}, t_{pw_z}$ Estimated (Millimetres)
STANDARD FACE				
Face model	(0.0,0.0,0.0)	(0.0,0.0,0.0)	(0,0,100)	(0,0,100)
Nose part	(0.0,0.0,0.0)	(0.0,0.0,0.0)	(0,0,100)	(0,0,100)
ROTATED FACE				
Face model	(1.57,0.0,0.0)	(1.57,0.0,0.0)	(0,0,100)	(0,0,100)
Nose part	(1.57,0.0,0.0)	(1.57,0.0,0.0)	(0,0,100)	(0,0,100)
TRANSLATED FACE				
Face model	(0.0,0.0,0.0)	(0.0,0.0,0.0)	(12,-4,102)	(12,-4,102)
Nose part	(0.0,0.0,0.0)	(0.0,0.0,0.0)	(12,-4,102)	(12,-4,102)

Table 7–4: Actual and Estimated Positions and Orientations

RESULTS FOR THE CLASS OF SYNTHETIC FACES				
Image	Rotation $R_{pw}(r,s,t)$ Actual (Radians)	Rotation $R_{pw}(r,s,t)$ Estimated (Radians)	Translation $t_{pw_x}, t_{pw_y}, t_{pw_z}$ Actual (Millimetres)	Translation $t_{pw_x}, t_{pw_y}, t_{pw_z}$ Estimated (Millimetres)
SCALED FACE				
Face model	(0.0,0.0,0.0)	(0.0,0.0,0.0)	(0,0,100)	(0,0,100)
Nose part	(0.0,0.0,0.0)	(0.0,0.0,0.0)	(0,0,100)	(0,0,100)
STRETCHED FACE				
Face model	(0.0,0.0,0.0)	(0.0,0.0,0.0)	(0,0,100)	(0,0,100)
Nose part	(0.0,0.0,0.0)	(0.0,0.0,0.0)	(0,0,100)	(0,0,100)
FACE WITH VARIATION				
Face model	(0.0,0.0,0.0)	(0.0,0.0,0.0)	(0,0,100)	(0,0,100)
Nose part	(0.0,0.0,0.0)	(0.0,0.0,0.0)	(0,0,100)	(0,0,100)

Table 7–5: Actual and Estimated Positions and Orientations

selections but six incorrect selections. These incorrect selections occured for two reasons. The majority were incorrect selections of a part with very large estimates of scale and stretch. These caused any remaining features of the selected model part to be predicted as outside the image and so the selection was verified. The remainder were incorrect selections of parts with only three surfaces. Since the selection process finds correspondences for three surfaces, if such a part has been selected the only action of the verification process is to establish the coordinate frame transformation and the verification is accepted.

Verification of the face model requires the identification of a fourth part in the image. Since the model is an exact representation of the object, prediction and verification of the missing part is straightforward. The nose part only contains three surfaces so no more surfaces are required to be found during verification.

The combined scale and stretch parameters for the face are recalculated following the acquisition of a fourth surfaces. The combined scale and stretch of both face and nose is, as expected, unity. The calculated rotation and translation of both parts is zero except for the expected translation along the Z–axis.

The second synthetic image contains the same basic face but this time rotated by $\pi/2$. Of the 21 selections originally made, the same eight as before are successfully verified. The six incorrect verifications are accepted for the same reasons as described earlier. This might not be the case if the image were not square and rotating the object caused the predicted positions to fall within the image. As before the nose requires only the calculation of the coordinate frame transformation for verification. The face requires requires a correspondence with a fourth surface before it is accepted.

The combined scale and stretch parameters are recalculated as unity. The rotation of both the nose part and the face model is determined to be the expected value of $\pi/2$.

The third of the synthetic images shows no scale, stretch or rotation but a small translation (in addition to the existing translation along the Z–axis) with components along all three axes. As before there are 21 selections, eight of which are accepted although only two are correct. The recalculated scale and stretch values are as expected, there is no rotation and the three components of the translation are correctly calculated.

The fourth image contains the basic face shape but this time it has been uniformly scaled by a factor of two. Of the original 18 selections only four pass the verification process. These include the two correct selections. After the verification process, the recalculated combined scale and stretch is, as expected, two with no rotation or translation estimated other than that along the Z–axis.

In the fifth image the basic face object has been stretched. The face object itself has been stretched by 0.8 and this value is used in the prediction and matching of a fourth surface during the verification process. The rotation of the object is zero as expected. The nose part is more interesting. Although originally stretched along the Y–axis by a factor of 1.1 because this deformation fell into the symmetric undetermined case described earlier then the parameters of scale and stretch could not be determined and an estimate was used instead. Since there are only three surfaces in this part, this estimate of the scale and stretch remains the only estimate available. In determining the coordinate frame transformation therefore these parameters are used to deform the model. Since they are only estimates this has a knock on effect with the estimation of the transformation. The rotation of the nose part is not the value of zero expected.

The last of the synthetic images exhibits individual variation in one of its surfaces. The surface is part of the face model and so the nose part is unaffected. During verification the remaining surface of the four in the face superpart is found and the parameters recalculated. Since there is only one surface showing variation, and that by only a very small amount, the overall effect of the variation on the calculated scale, stretch, rotation and translation for the model is minimal.

The Class of Screwdrivers

For all of the images containing just one screwdriver there was just one selection made, the correct selection of the screwdriver handle part. Since this was the only selection made and this part consists of only three surfaces, the only action of the verification process is to establish the coordinate frame transformation between deformed model and the object. The results of part verification for the screwdrivers is shown in Tables (7–6), (7–7) and (7–8).

RESULTS FOR THE CLASS OF SCREWDRIVERS

Image	Number of Selections	Subset Selections	Selections Verified	Incorrect Verifications	Correct Verifications
Small Screwdriver	1	0	1	0	1
Medium Screwdriver	1	0	1	0	1
Large Screwdriver	1	0	1	0	1
Scaled Screwdriver	1	0	1	0	1
Multiple Screwdrivers	2	0	2	0	2

Table 7–6: Summary of the Verifications Made During Model Part Verification

As discussed in the previous chapter the segmentation of the surface data of the small screwdriver is rather poor. This resulted in a rather poor estimate of the combined scale and stretch. Since there are only three surfaces in this part there is no opportunity to recalculate the data more accurately following a further correspondence. As a result, the estimate of rotation between deformed model and screwdriver is also poor.

For the remaining images where the segmentation is better the estimates of the transformation are good. This shows that even with the barest minimum of information the part selection and part verification processes can return reasonable values if the information is reliable.

The last image produced two selections. Each of these is a correct selection of the screwdriver handle part. The estimates of the scale and stretch are unusually good and this is reflected in the estimates of the coordinate frame transformation.

The Class of Human Faces

RESULTS FOR THE CLASS OF SCREWDRIVERS						
Image	S_{gx} Actual	S_{gx} Estim.	S_{gy} Actual	S_{gy} Estim.	S_{gz} Actual	S_{gz} Estim.
SMALL SCREWDRIVER Handle part	0.538	0.543	N/A	N/A	1.550	2.765
MEDIUM SCREWDRIVER Handle part	1.000	0.894	N/A	N/A	1.000	0.992
LARGE SCREWDRIVER Handle part	1.188	1.135	N/A	N/A	1.150	1.178
SCALED SCREWDRIVER Handle part	2.000	1.788	N/A	N/A	2.000	1.984
MULTIPLE SCREWDRIVERS Standard handle part	1.000	0.968	N/A	N/A	1.000	1.132
Large handle part	1.188	1.295	N/A	N/A	1.150	1.224

Table 7–7: The Deformation Parameters Estimated During Model Part Verification

RESULTS FOR THE CLASS OF SCREWDRIVERS				
Image	Rotation	Rotation	Translation	Translation
	$R_{pw}(r,s,t)$	$R_{pw}(r,s,t)$	$t_{pw_x}, t_{pw_y}, t_{pw_z}$	$t_{pw_x}, t_{pw_y}, t_{pw_z}$
	Actual	Estimated	Actual	Estimated
	(Radians)	(Radians)	(Millimetres)	(Millimetres)
SMALL SCREWDRIVER Handle part	(1.6,0.0,0.0)	(2.0,1.0,1.6)	(7,-11,972)	(7,-11,972)
MEDIUM SCREWDRIVER Handle part	(0.0,0.6,1.6)	(0.1,0.5,1.6)	(-9,-2,983)	(-9,-2,983)
LARGE SCREWDRIVER Handle part	(0.0,0.2,1.8)	(0.0,0.1,1.8)	(-25,-6,980)	(-25,-6,980)
SCALED SCREWDRIVER Handle part	(0.0,0.6,1.6)	(0.1,0.5,1.6)	(-18,-4,991)	(-19,-4,991)
MULTIPLE SCREWDRIVERS Medium handle part	(0.0,0.5,1.6)	(0.1,0.4,1.7)	(6,-3,983)	(6,-3,983)
Large handle part	(0.0,0.2,1.8)	(0.1,0.4,1.6)	(-21,36,980)	(-21,36,980)

Table 7–8: Actual and Estimated Positions and Orientations

RESULTS FOR THE CLASS OF HUMAN FACES					
Image	Number of Selections	Subset Selections	Selections Verified	Incorrect Verifications	Correct Verifications
Adult's Face 1	113	44	18	15	3
Adult's Face 2	12	0	6	4	2
Child's Face	47	12	15	12	3

Table 7–9: Summary of Verifications Made During Model Part Verification

RESULTS FOR THE CLASS OF HUMAN FACES						
Image	S_{gx} Actual	S_{gx} Estim.	S_{gy} Actual	S_{gy} Estim.	S_{gz} Actual	S_{gz} Estim.
ADULT'S FACE 1						
Nose part	1.000	1.047	1.000	1.099	1.000	1.123
Eyes part	1.000	1.203	1.000	1.062	1.000	1.201
Face model	1.000	1.091	1.000	1.100	1.000	1.167
ADULT'S FACE 2						
Nose part	1.000	0.989	1.000	1.106	1.000	0.992
Face model	1.000	1.003	1.000	0.902	1.000	0.980
CHILD'S FACE						
Nose part	1.000	0.843	1.000	0.824	1.000	0.953
Eyes part	1.000	0.920	1.000	0.799	1.000	0.805
Face model	1.000	0.811	1.000	0.912	1.000	0.822

Table 7–10: The Deformation Parameters Estimated During Model Part Verification

RESULTS FOR THE CLASS OF HUMAN FACES				
Image	Rotation	Rotation	Translation	Translation
	$R_{pw}(r,s,t)$	$R_{pw}(r,s,t)$	$t_{pw_x}, t_{pw_y}, t_{pw_z}$	$t_{pw_x}, t_{pw_y}, t_{pw_z}$
	Actual	Estimated	Actual	Estimated
	(Radians)	(Radians)	(Millimetres)	(Millimetres)
ADULT'S FACE 1				
Nose part	(0.0,0.0,0.0)	(0.1,0.3,1.6)	(-65,-10,103)	(-67,-11,107)
Eyes part	(0.0,0.0,0.0)	(0.2,-0.2,0.8)	(-65,21,140)	(-64,-23,142)
Face superpart	(0.0,0.0,0.0)	(-0.1,0.2,1.2)	(-65,-41,120)	(-65,-40,120)
ADULT'S FACE 2				
Nose part	(0.0,0.0,0.0)	(0.0,0.5,1.4)	(0,0,130)	(7,-5,132)
Face superpart	(0.0,0.0,0.0)	(0.0,0.4,0.5)	(0,-31,147)	(0,-28,150)
CHILD'S FACE				
Nose part	(0.0,0.0,0.0)	(-0.1,0.1,0.6)	(-64,-10,103)	(-63,-12,102)
Eyes part	(0.0,0.0,0.0)	(0.2,0.2,1.1)	(-64,-21,140)	(-63,-19,139)
Face superpart	(0.0,0.0,0.0)	(0.0,-0.2,1.1)	(-64,-41,120)	(-64,-42,131)

Table 7–10a: Actual and Estimated Positions and Orientations

The class of objects used here are more complex than those dealt with so far. At this stage however the part verification process seeks only to verify the presence of the different parts as selected.

The first face image is that which the class model is based on. Needless to say, although the model is not an exact representation, it is close, and this is reflected in the selections made and the accuracy of the deformation parameters. Because of the large number of surfaces, both represented in the model and visible in the image there are a large number of redundant selections. Once the first selection of this set is verified during which all the remaining correspondences are found then the redundant selections are quickly identified as subsets and eliminated without verification. Unlike the previous examples, most of the selections incorrectly verified are verified by finding possible (though poor) correspondences from amongst the large number of surfaces in the image. The eyes and nose parts are successfully verifed with good estimates of both the combined scale and stretch and coordinate frame transformation.

The results obtained from the second image are not as good. The original selections are fewer as there are less object features visible in the image. There are no redundant selections at all. However, both parts and the superpart itself are all selected and passed to the verification process. The face model surfaces are verified, the surfaces representing the area beneath the lips are predicted as lying outside the image. The nose part is also verified, the segmentation of the nose on this face being particularly good. The problem with this image arises when trying to verify the eyes part. The verification process attempts to find correspondences for *all* the surfaces represented. However, in this image there are no surfaces segmented for the eyeballs and since the verification process cannot explain their absence it fails and the original selection is rejected. This shortcoming of the verification process is discussed in the conclusions and further work suggested to overcome it.

The segmentation of the third image is similar to that of the first image and the selections made reflect this. The reduced number of selections originally made is due to the shape of forehead being estimated as hyperbolic instead of the positive elliptical surface represented in the model. This represents no problem to the verification process however, which does not identity an incorrect match on the grounds of shape. The correct correspondence for the forehead is found by verifying a selection made using an alternative combination of surfaces

7.3 Superpart (or Model) Verification

7.3.1 Theory

The intention of the superpart (or model) verification process is to verify the existence in the image of a superpart or model selected by the superpart selection process. Selection of a superpart or model is made because of the existence in the image of some of its constituent surfaces, parts or superparts. Verification is achieved by identifying correspondences for all remaining surfaces, parts and superparts represented in the superpart or model. To avoid confusion only the case where a selected superpart consists entirely of parts will be covered here. Verification by the identification of surfaces was covered in the previous section and the verification of a selected model by its superparts is essentially the same as the process described here.

Generating Model to Object Correspondences

When one or more parts of an object have been identified in the image then the existence in the image of the superpart to which the parts belong must be verified. The problem is similar to that of part verification but with some important differences. To verify the existence of a superpart, correspondences for all the parts represented in the model superpart must be established. However, an object part is not immediately visible in the image. The part verification process will have identified several object parts and these may provide correct matches. Failing this though, the deformation, position and orientation of any missing part will have to be estimated and the part verification process used in a *top down* manner to search for and identify the visible component surfaces of that part in the image.

Reducing the Number of Matches

Generally the number of parts and surfaces in a superpart is small resulting in a small search space. However, for efficiency this space should be pruned in the same manner as that used for part verification taking advantage of the more accurately calculated parameters of the superpart.

7.3.2 Implementation

Once again a search tree is used to represent the possible combinations of matches of model parts with object parts already identified in the image. The width of the tree represents the number of object parts available while the depth of the tree represents the number of parts in the model. Because of the smaller number of parts than surfaces, the initial superpart verification tree is small compared to those used in part verification. However, further trees may be generated if the verification process uses the *top down* approach to identify parts, such as those containing only one or two surfaces, not found during the part selection and verification processes.

Starting Superpart Verification

The superpart verification process takes as input a list of combinations of model parts whose corresponding object parts are believed to exist in the image. The superpart or model to be verified is that to which these model parts belong.

How a superpart is to be verified depends upon just how much evidence is available in the original selection. If only one or two parts were used to select the superpart then an approximation of the combined global scale and stretch parameters of the superpart is obtained from the scale and stretch parameters of the part (or parts) transformed into the superpart coordinate frame using a least squares fit if appropriate.

Estimating the Deformation of the Object Superpart

In making the selection, the superpart selection process has estimated the combined scale and stretch parameters $S_{gx_{sp}}$, $S_{gy_{sp}}$ and $S_{gz_{sp}}$ of the superpart sp according to the number of correspondences available. An estimate of the global scale S_g will also have been made. As new correspondences are achieved then the estimate of the combined scale and stretch and the estimate of the global scale may both be updated.

The method by which the combined scale and stretch estimate is updated depends on the number of correspondences available. Three or more surfaces are ideal as this enables the combined scale and stretch of the superpart to be determined independently of the scale, stretch and coordinate frame transformations of its parts. Where four correspondences are identified then the problem becomes overspecified and can be solved by the least–squares fit method described previously.

Estimating the Rotation of the Deformed Superpart

The way in which the rotation between the superpart coordinate frame and that of the world $R_{spw}(r, s, t)$ is initially calculated also depends on how many model to object part correspondences were in the original selection. With a single correspondence then the rotation of the superpart is estimated from the rotation of the part using the geometric relationship between the two as specified in the model. A similar approach is used where there are two correspondences but since each part could, and normally will, give a different estimate a least squares fit is used to determine a rotation that fits best. Three correspondences are ideal because each set of three positions yields a set of two vectors as in Figure (7–2). This enables the rotation of the superpart to be determined independently of the rotation of its parts and any errors they may contain.

If further model to object correspondences are established then the rotation is recalculated using the method appropriate to the number of correspondences. Where there are four or more correspondences between the model and object then the problem becomes overspecified and can be solved as before using the least–squares fit detailed in Section 8.4.1. The inverse rotation between the coordinate frame of the world and that of the superpart $R_{wsp}(R', S', T')$ is determined at the same time.

Estimating the Translation of the Deformed Superpart

The translation of the superpart is determined in a similar way to the rotation. Where the selection process has made a selection based on only one or two correspondences then the translation is estimated using the relationship between part and superpart as specified in the model. Any errors in the calculated position of the part or parts will be propogated into the calculated position of the superpart. Three correspondences enables the translation to be determined independently.

If any new correspondences are found then the method appropriate to the number of correspondences is used to update the estimate.

Estimating the Individual Variation of the Deformed Superpart

Estimation of the individual variations in the position and orientations of the parts within the superpart can only be calculated where there are correspondences between two or more parts. The individual variation is not normally used in the superpart verification process unless a model contains two, or more, identical parts.

Estimating the Deformation of the Component Parts

The deformation of a component part has been expressed in terms of the deformation relative to the original model part S_{gx_p}, S_{gy_p} and S_{gz_p}. These parameters represent the combined effects of not only the global scale but the part stretch at each level in the hierarchy of object parts and superparts. If the combined stretch parameters of the superpart are known, then assuming they are accurate, any difference between these and the combined parameters of its component parts must be due solely to the stretch of the parts, subject to the geometric relationship between them. The stretch parameters of the part S_x, S_y and S_z as defined in Section 3.3.2 can be resolved dependent on the accuracy to with which the combined scale and stretch parameters of the superpart are known.

Part Prediction

Having determined the combined scale and stretch of the model superpart and the coordinate frame transformation between the deformed model and the world, the position and orientation of any parts in the model superpart for which corresponding object parts have not yet been identified can be predicted. The combined scale and stretch of the superpart is used as an estimate of the combined scale and stretch of the component part, the components of the deformation being transformed into the coordinate frame of the part.

Identification of a Correct Correspondence

The degree to which an object part fits that predicted from the model is also determined on the basis of how closely its position matches that predicted. The degree of fit is evaluated as in equation (7.3) and a correct match assumed if the difference falls within the threshold T.

Expanding the Search Tree

For each object part whose position is sufficiently close to that predicted, the correspondence is assumed to be correct and the deformation and coordinate frame transformation updated accordingly. Those nodes in the search tree which represent a correct correspondence are tagged to be expanded further, all other nodes are discarded. Where no corresponding object part is found, the deformation, orientation and position parameters of the part calculated earlier are used to call the part verification process to search the image for the appropriate surfaces. If the part verification process is able to identify the part or explain its absence then the search continues ; otherwise it terminates and the superpart selection is deleted.

Once the existence of a superpart has been verified then as before the list of superpart selections is searched and any selections in which the selecting parts are a *subset* of those in the object part just verified are deleted from further examination. Since the list of selections is ranked according to the number of correspondences used

RESULTS FOR THE CLASS OF SYNTHETIC FACES		
Image	S_g True	S_g Estim.
STANDARD FACE		
Face model	1.000	1.000
ROTATED FACE		
Face model	1.000	1.000
TRANSLATED FACE		
Face model	1.000	1.000

Table 7–11: The Global Scale Estimated During Superpart Verification

in making the selection, this means that any redundant selections are eliminated by those selections based on greater numbers of correspondences. The global scale and stretch parameters, whether combined or separated, are transformed into the coordinate frame of the superpart or model of which this superpart is part.

The verified superparts, but not any verified models, are grouped according to the superparts or models they form part of to be passed to the superpart (model) selection process. Control passes between these two process at successively higher levels in the model hierarchies until finally there are no more superparts left to be verified and the recognition process terminates.

7.3.3 Results

Three further sets of results are presented here. The first set of results were obtained using the set of synthetically generated images to test the various aspects of the model superpart verification process. The second and third sets are from classes of real objects.

The Class of Synthetic Faces

To test the various features of the superpart selection process the set of synthetic images defined earlier were used.

RESULTS FOR THE CLASS OF SYNTHETIC FACES		
Image	S_g True	S_g Estim.
SCALED FACE		
Face model	2.000	2.000
STRETCHED FACE		
Face model	1.000	1.000
FACE WITH VARIATION		
Face model	1.000	1.000

Table 7–12: The Global Scale Estimated During Superpart Verification

RESULTS FOR THE CLASS OF SYNTHETIC FACES						
Image	S_x True	S_x Estim.	S_y True	S_y Estim.	S_z True	S_z Estim.
STANDARD FACE						
Face model	1.000	1.000	1.000	1.000	1.000	1.000
Nose part	1.000	1.000	1.000	1.000	1.000	1.000
ROTATED FACE						
Face model	1.000	1.000	1.000	1.000	1.000	1.000
Nose part	1.000	1.000	1.000	1.000	1.000	1.000
TRANSLATED FACE						
Face model	1.000	1.000	1.000	1.000	1.000	1.000
Nose part	1.000	1.000	1.000	1.000	1.000	1.000

Table 7–13: The Deformation Parameters Estimated During Superpart Verification

RESULTS FOR THE CLASS OF SYNTHETIC FACES						
Image	S_x	S_x	S_y	S_y	S_z	S_z
	True	Estim.	True	Estim.	True	Estim.
SCALED FACE						
Face model	1.000	1.000	1.000	1.000	1.000	1.000
Nose part	1.000	1.000	1.000	1.000	1.000	1.000
STRETCHED FACE						
Face model	1.000	1.000	0.800	0.800	1.000	1.000
Nose part	1.000	1.000	1.375	1.348	1.000	1.078
FACE WITH VARIATION						
Face model	1.000	1.000	1.000	1.010	1.000	1.050
Nose part	1.000	1.000	1.000	0.990	1.000	0.952

Table 7–14: The Deformation Parameters Estimated During Superpart Verification

For the first synthetic image the superpart selection process has identified the face object and the nose part as consistent with belonging to the same object. These two sets of features constitute all that is represented in the corresponding model so that no further features need be searched for. All that remains for the superpart verification process to do is to recalculate the coordinate transformation and the estimates of deformation for the top level of the object. There is no deformation of the object and so all the scale and stretch parameters are unity. The final estimate of the combined scale and stretch of the object is now passed back down through the model hierarchy to resolve the scale and stretch of the constituent parts or, in this case, part. However, since there is no deformation detailing, this process at this stage is pointless. A full description is given later. The final estimate of the transformation is that expected showing no rotation or translation other than that along the Z–axis. There is no deformation of the object and so all the scale and stretch parameters are unity.

The second synthetic image is much the same as the first apart from the rotation. The function of the superpart verification process is merely to recalculate the transformation between object and model and to recalculate the top level deformation. The rotation

RESULTS FOR THE CLASS OF SYNTHETIC FACES				
Image	Rotation	Rotation	Translation	Translation
	$R_{spw}(r,s,t)$	$R_{spw}(r,s,t)$	$t_{spw_x}, t_{spw_y}, t_{spw_z}$	$t_{spw_x}, t_{spw_y}, t_{spw_z}$
	Actual	Estimated	Actual	Estimated
	(Radians)	(Radians)	(Millimetres)	(Millimetres)
STANDARD FACE				
Face model	(0.0,0.0,0.0)	(0.0,0.0,0.0)	(0,0,100)	(0,0,100)
ROTATED FACE				
Face model	(1.57,0.0,0.0)	(1.57,0.0,0.0)	(0,0,100)	(0,0,100)
TRANSLATED FACE				
Face model	(0.0,0.0,0.0)	(0.0,0.0,0.0)	(12,-4,102)	(12,-4,102)
SCALED FACE				
Face model	(0.0,0.0,0.0)	(0.0,0.0,0.0)	(0,0,100)	(0,0,100)
STRETCHED FACE				
Face model	(0.0,0.0,0.0)	(0.0,0.0,0.0)	(0,0,100)	(0,0,100)
FACE WITH VARIATION				
Face model	(0.0,0.0,0.0)	(0.0,0.0,0.0)	(0,0,100)	(0,0,100)

Table 7–15: Actual and Estimated Positions and Orientations

is the expected value of $\pi/2$ and the scale and stretch is evaluated as unity. This value is passed down to the lower parts of the hierarchy but since there is no deformation to resolve in this image, the process is not detailed here.

The action of the process for the translated face in the third image is the same as for the rotated face except that a translation. rather than a rotation, is estimated.

The fourth image contains the scaled up version of the basic face. Once again the action of the superpart verification process is merely to recalculate the coordinate transformation and the deformation parameters. The coordinate transformation is as expected but the combined scale and stretch parameters of the object are all 2.0. Using the criteria defined earlier regarding the resolution of scale and stretch these combined scale and stretch parameters should be explained as far as possible by a change in scale. In fact in this case they can be explained completely by a scale change so that the global scale S_g becomes 2.0. The combined scale and stretch of the object is now resolved by dividing through by the scale to give an object stretch of unity. The scale and stretch of the face object are now passed down to the nose part. The coordinate frames of the part and that of the object are aligned so the stretch of the part S_{xnose}, $S_{y_{nose}}$ and S_{znose} is given by:

$$S_{xnose} = \frac{S_{gx_{nose}}}{S_g S_{x\,face}} \tag{7.4}$$

$$S_{y_{nose}} = \frac{S_{gy_{nose}}}{S_g S_{y\,face}} \tag{7.5}$$

$$S_{znose} = \frac{S_{gz_{nose}}}{S_g S_{z\,face}} \tag{7.6}$$

This determines the stretch of the nose part to also be unity.

The stretched face in the fifth image uses the same strategy. Following the recalculation of the coordinate frame transformation the combined scale and stretch of the face object is determined. The global scale is estimated and this used to resolve the stretch of the face object. The global scale and the stretch of the object are passed down to the nose part to enable the stretch of the part to be determined.

Exactly the same thing happens for the sixth image where the basic face shows variation though the degree of scale and stretch to be dealt with is far less.

RESULTS FOR THE CLASS OF SCREWDRIVERS		
Image	S_g Actual	S_g Estim.
SMALL SCREWDRIVER		
Screwdriver Model	0.873	1.225
MEDIUM SCREWDRIVER		
Screwdriver Model	1.000	0.942
LARGE SCREWDRIVER		
Screwdriver Model	1.264	1.156
SCALED SCREWDRIVER		
Screwdriver Model	2.000	1.883
MULTIPLE SCREWDRIVERS		
Screwdriver Model	1.000	1.047
Screwdriver Model	1.264	1.259

Table 7–16: The Global Deformation Calculated During Superpart Verification

The Class of Screwdrivers

In all the images of screwdrivers the only part with three or more surfaces that appears in the image is the screwdriver handle part. As has been described earlier the existence of this part of three surfaces is sufficient to give an estimate of the combined scale and stretch and determine the coordinate transformation that maps deformed model to object. The existence of a single part is the minimum required to select a superpart as happens here. In each of the images the screwdriver model is selected on the basis of the verified presence of the screwdriver handle. Because there is no other data to base the selection on, the deformation of the screwdriver is thought to be the same as that

RESULTS FOR THE CLASS OF SCREWDRIVERS						
Image	S_x	S_x	S_y	S_y	S_z	S_z
	Actual	Estim.	Actual	Estim.	Actual	Estim.
SMALL SCREWDRIVER						
Screwdriver model	0.649	0.439	N/A	N/A	1.409	2.257
Handle part	0.950	1.000	N/A	N/A	1.260	1.000
MEDIUM SCREWDRIVER						
Screwdriver model	1.000	0.949	N/A	N/A	1.000	1.053
Handle part	1.000	1.000	N/A	N/A	1.000	1.000
LARGE SCREWDRIVER						
Screwdriver model	1.194	0.981	N/A	N/A	0.981	1.019
Handle part	0.762	1.000	N/A	N/A	0.958	1.000
SCALED SCREWDRIVER						
Screwdriver model	1.000	0.949	N/A	N/A	1.000	1.053
Handle part	1.000	1.000	N/A	N/A	1.000	1.000
MULTIPLE SCREWDRIVERS						
Standard						
Screwdriver model	1.000	0.924	N/A	N/A	1.000	1.081
Handle part	1.000	1.000	N/A	N/A	1.000	1.000
Large						
Screwdriver model	1.194	1.029	N/A	N/A	0.981	0.972
Handle part	0.762	1.000	N/A	N/A	0.958	1.000

Table 7–17: The Deformation Parameters Estimated During Superpart Verification

RESULTS FOR THE CLASS OF SCREWDRIVERS				
Image	Rotation	Rotation	Translation	Translation
	$R_{spw}(r, s, t)$	$R_{spw}(r, s, t)$	$t_{spw_x}, t_{spw_y}, t_{spw_z}$	$t_{spw_x}, t_{spw_y}, t_{spw_z}$
	Actual	Estim.	Actual	Estim.
	(Radians)	(Radians)	(Millimetres)	(Millimetres)
SMALL				
SCREWDRIVER				
Screwdriver model	(1.6,0.0,0.0)	(2.0,1.0,1.6)	(7,-11,972)	(7,-11,972)
MEDIUM				
SCREWDRIVER				
Screwdriver model	(0.0,0.6,1.6)	(0.1,0.5,1.6)	(-9,-2,983)	(-9,-2,983)
LARGE				
SCREWDRIVER				
Screwdriver model	(0.0,0.2,1.8)	(0.0,0.1,1.8)	(-25,-6,980)	(-25,-6,980)
SCALED				
SCREWDRIVER				
Screwdriver model	(0.0,0.6,1.60)	(0.1,0.5,1.6)	(-18,-4,991)	(-19,-4,991)
MULTIPLE				
SCREWDRIVERS				
Medium				
screwdriver model	(0.0,0.5,1.6)	(0.1,0.4,1.7)	(6,-3,983)	(6,-3,983)
Large				
screwdriver model	(0.0,0.2,1.8)	(0.1,0.4,1.6)	(-21,36,980)	(-21,36,980)

Table 7–18: Actual and Estimated Positions and Orientations

of the part making the selection. Apart from the screwdriver handle, the screwdriver model consists of another part, the screwdriver shaft. This consists of only one surface and so will not have been identified by the part selection process. The search for a part is now driven in a top down manner, the selected model and the best estimate of its deformation used to direct the search for the unfound part. The accuracy with which the position of the part is predicted naturally depends on the accuracy with which the deformation parameters of the screwdriver are known.

For the image of the small screwdriver, the poor segmentation of the range data led to an inaccurate estimation of the combined scale and stretch during part selection. In the absence of any further data this is the estimate still being used and it still causes problems. However, the distance between the position of the shaft surface and its predicted position is within the threshold allowed (just) and the correct correspondence is achieved.

For the medium screwdriver the same process applies although, because the original estimate of scale and stretch was better, the predicted position of the surfaces is much closer to the actual position.

The same process applies to the image of the large screwdriver, the image of the scaled screwdriver and to each of the screwdrivers in the image of multiple screwdrivers.

Although the shaft is a part, it only consists of one surface. This means that it contributes nothing to the estimation of the deformation parameters and the transformation of the object as a whole. The combined scale and stretch of the object is considered to be that obtained from the handle. The combined scale and stretch is divided through by the global scale, and the resolved scale and stretch passed back down to the handle part. This is a far from satisfactory arrangement and further work is continuing on this.

The Class of Human Faces

RESULTS FOR THE CLASS OF HUMAN FACES		
Image	S_g Actual	S_g Estim.
ADULT'S FACE 1		
Face model	1.000	1.120
ADULT'S FACE 2		
Face model	1.000	0.994
CHILD'S FACE		
Face model	1.000	0.852

Table 7–19: The Global Deformation Calculated During Superpart Verification

RESULTS FOR THE CLASS OF HUMAN FACES						
Image	S_x Actual	S_x Estim.	S_y Actual	S_y Estim.	S_z Actual	S_z Estim.
ADULT'S FACE 1						
Face model	1.000	0.935	1.000	0.981	1.000	1.003
Nose part	1.000	1.149	1.000	0.966	1.000	1.069
Eyes part	1.000	1.042	1.000	1.001	1.000	1.039
CHILD'S FACE						
Face model	1.000	0.989	1.000	0.967	1.000	1.119
Nose part	1.000	1.091	1.000	0.970	1.000	0.845
Eyes part	1.000	0.962	1.000	1.107	1.000	0.862

Table 7–20: The Deformation Parameters Estimated During Superpart Verification

RESULTS FOR THE CLASS OF HUMAN FACES				
Image	Rotation $R_{spw}(r,s,t)$ Actual (Radians)	Rotation $R_{spw}(r,s,t)$ Estimated (Radians)	Translation $t_{spw_x}, t_{spw_y}, t_{spw_z}$ Actual (Millimetres)	Translation $t_{spw_x}, t_{spw_y}, t_{spw_z}$ Estimated (Millimetres)
ADULT'S FACE 1				
Face model	(0.0,0.0,0.0)	(-0.1,0.2,1.2)	(-65,-41,120)	(-65,-40,120)
CHILD'S FACE				
Face model	(0.0,0.0,0.0)	(0.0,-0.2,1.1)	(-64,-41,120)	(-64,-42,131)

Table 7–21: Actual and Estimated Positions and Orientations

Where the objects are more complex with more parts available to make a selection the effect of a poor estimate of deformation or position has a less pronounced effect on the determination of the parameters for the whole object.

The first adult face model is selected by the existence of the nose part, the eyes part and the surfaces of the model itself. There are no further features left to find correspondences for and so the action of the superpart verification process is merely to update the deformation and transformation parameters. With so many sources of independent information a reliable estimate of the global estimate can be made. There is enough evidence available for the combined scale and stretch of the object to be determined independently of its parts. These values can then be accurately resolved by dividing them by the value for the global scale calculated previously. The coordinate frames of both the nose and the eyes parts are aligned with the coordinate frame of the face so that the global scale and the stretch of the face can simply be passed down the hierarchy to resolve the stretch of both of these parts.

The second adult face presents something of a problem. The face model is selected on the basis of the nose part and the surfaces of the face itself. Superpart verification seeks to verify the existence of the object by finding any parts without correspondences, in this case the eyes part. The recognition process, as when seeking a correspondence for the screwdriver shaft, switches to a top down approach and uses the models parameters to direct the search for the missing part. The search for the part goes well with most of the surfaces being correctly identified. However, no surfaces are identified as representing the eyeballs and so the search for the nose part fails. This results in the verification of the face object itself being rejected and the original selection is deleted.

Because of the similarities in structure between the first adults face and the childs face the action of the superpart selection process is similar to that for the first image. Selection is made on the basis of the nose and eyes parts and of the surfaces of the face itself. With all the features identifed it just remains for the coordinate frame transformation to be identified. The final value for the global scale is evaluated as significantly less than one indicating that the face is smaller overall than the adult's face. This value for the global scale and the stretch parameters of the object itself are passed back down to the two parts. When the combined scale and stretch of the two parts is resolved into the part stretch, values of near unity are achieved. This does not mean that the parts have the same stretch as those in the adults face but that there is little stretch of the parts *relative* to the scaled and stretched face object.

7.4 Comparison with Previous Work

Those systems that use interpretation tree searches [Gaston & Lozano-Pérez, 1984], [Grimson & Lozano-Pérez, 1984], [Drumheller, 1987], [Pollard *et al*, 1987], [Ikeuchi, 1987], [Lozano-Pérez *et al*, 1987], [Knapman, 1987], [Simsarian *et al*, 1990], and [Grimson, 1990b] usually find interpretations for quite large numbers of object features. Because the "local" constraints that are applied to reduce the search space use only locally available data while the individual pairings in an interpretation might all be correct, the interpretation might not form a consistent whole. The normal solution to this is that once each interpretation tree has been fully expanded, to verify each feasible interpretation by trying to find a transformation that consistently maps the model features to those of the object (or the inverse transformation from object to model). If there are a large number of interpretations or a large number of correspondences in each the geometric computations required can be quite expensive.

In the work described here, the interpretations produced by the model selection process include only three object features even though an object may have many more than that. Some of the interpretations will consist entirely of correct matches. If these can be found using local constraints then there is no longer the need to check the interpretations geometrically. By finding the consistent transformation early on, the global geometric constraints can be applied to guide the search for any object features not included in the original interpretation.

The search tree described here is similar to that used in [Goad, 1985]. In that work the identity of the object was known in advance but its position in the image and the correspondence between the features in the model and those in the object were not. Here, while the identity of the object is not known with certainty, during the verification process the identity of the object is assumed to be that hypothesised during part selection – unless the verification process proves otherwise. In Goad's work object features were also searched for one by one, and the current estimate of the object's orientation and position in the image updated as new correspondences were found.

A similar approach to the one described here was developed in [Huttenlocher & Ullman, 1987]. A small number of model features were searched for in the image and used not only to establish the transformation between object and model but a scale factor as well. With the transformation known the search for any remaining features could be guided to avoid the combinatorial explosion problem. The objects to be identified were laminar, lying on a flat surface and so the features as modelled could, in the absence of occlusion, always be found in the image. For three dimensional objects some features are usually self occluded and so the existence in the image of any particular model feature cannot be guaranteed. The proposed extension for three dimensional objects was to model a series of views from a large number of different orientations effectively reducing the problem to a series of two dimensional matches. In the work described here the overall philosophy is the same but instead of initially seeking model to object correspondences, object to model correspondences are sought for the objects in the image since these object features obviously are visible whatever the orientation of the object!

7.5 Conclusion

7.5.1 Summary

Two processes have been described here. The first, that of part verification, takes as input the object to model surface correspondences provided by the part selection process and uses them to establish an estimate of the combined global scale and stretch parameters of the part and the orientation and position of the model part coordinate frame. These geometric constraints are used to reduce the size of the search tree search for any surfaces in the model part that remain unmatched. The coordinate frame transformation allows the degree to which a surface in the image fits a surface in the model to be estimated and any image surface whose degree of fit falls outside some threshold may be disregarded from the matching process. Since any further matching using this surface would also be invalid the search tree can be pruned at this point reducing the size of the tree and therefore the number of nodes that need to be traversed. Where the degree of fit between an image and a model surface is within some threshold, the addition of another surface enables the geometric constraints to be refined and the shape, orientation and position of any remaining unmatched model surface to be predicted with greater accuracy.

The second process, of superpart (or model) verification works in a similar way but searches for surfaces, parts or superparts to verify the existence of an object superpart of object in the image. Because a part is not immediately identifiable in the image, if the part has not already been identified elsewhere then the verification process switches to a top down approach to search the image for the part's surfaces.

7.5.2 The Efficiency of Model Verification

If an object, corresponding to a model p which is represented by m_p features, is thought to exist in an image of I surfaces, then since 3 object to model correspondences are already known the maximum number of possible verified matches is given by:

$$number\ of\ verified\ matches = I^{(m_p-3)} \tag{7.7}$$

At *worst* a verification will be required for each selection made and so combining Equations (7.7) and (6.5) the maximum possible number of verified matches is given by:

$$total\ number\ of\ verified\ matches = \frac{I!}{(I-3)!3!}\sum_{p=1}^{M}\frac{m_p!}{(m_p-3)!}I^{(m_p-3)} \tag{7.8}$$

If I is much larger than 3 then the number of combinations of object surfaces is approximately I^3. Equation (7.8) now becomes:

$$total\ number\ of\ verified\ matches \approx \sum_{p=1}^{M}\frac{m_p!}{(m_p-3)!}I^{m_p} \tag{7.9}$$

The maximum number of verified matches is therefore an exponential function of the number of features in a model. For a given image the number of features in the model can reasonably be expected to remain constant reducing the function to that of a polynomial. Since the performance of the model selection process is not dependent on the number of objects in the image, if both objects and models are broken down into conceptual parts (as here) then the number of features in each model m_p is reduced at the expense of a larger number of models M. This will reduce the maximum number of possible verified matches though there is now the added cost of verifying object superparts on the basis of identified parts.

It should be stressed that this figure is the *maximum* number of verified matches that could possibly occur. As the results have shown, the effects of local constraints during part selection and the use of global geometric constraints and subset elimination during verification will normally reduce this figure substantially.

7.5.3 Verification of the Object

The method has been tested with both synthetic and real range data images and shown to work. Where the initial model selection was correct the search tree search will generally match the remaining model surfaces with their corresponding surfaces in the image and accurately estimate the combined global scale and stretch parameters of the model part and the orientation and position of the local part coordinate frame in the world coordinate frame.

Failure to successfully verify a part that was correctly selected occurred most often when one or more of the three image to model correspondences provided by the part selection process shows a large variation in position. Since at the selection stage any differences between the configuration of the image surfaces and those of the model is considered to be due entirely to the effects of scale and stretch, this leads to scale/stretch values that are significantly different than the actual values. This causes poor shape and position prediction and results in either an incorrect match or a failure to match the model surface with any of the corresponding object surfaces.

More rarely, the part verification process verifies the existence of an *incorrect* model part selection. This occurs where the configuration of the three image surfaces used in the initial selection yield geometric constraints such that all the remaining unmatched model surfaces are either predicted to be facing away from the world or (more usually) outside the boundaries of the image and therefore not to be searched for in the image. When this happens the incorrectly selected part will be verified on the strength of the three correspondences used in the selection and passed to the superpart selection process.

The superpart model selection process uses the configuration of the parts verified in the part verification process to select superparts or models. Since the confidence in the existence of the parts used to make the selection is high and the geometric parameters are accurately known by this stage, then the search tree produced can be heavily pruned, often to the minimum possible.

Incorrect superpart selections are quite common for the reasons given earlier. However, with the more accurate constraints available in the superpart verification process and the reduced amount of data to be matched with the model, incorrect superpart verifications are rare.

7.5.4 Estimating the Parameters of Deformation

Provided that an object was segmented reasonably such that at least three of the segmented object surfaces were in a recognisable configuration the part selection process could obtain at least one fair estimate of the combined scale and stretch of the object part. This was generally sufficient to guide the verification process and enable the remaining deformation parameters to be resolved.

Chapter 8

Numerical Methods

8.1 Introduction

In this chapter the numerical algorithms used during the model selection and model verification processes are described in detail. The algorithms are loosely divided into four groups; those used in part selection, those used in superpart (or model) selection, those used in part verification and those used in superpart (or model) verification. The algorithms are only loosely grouped, many being used by more than one process.

The emphasis when designing the algorithms used in model selection was for fast and efficient identification of object parts. As a result, the methods developed use relatively simple constraints that are based on locally available data and are independent of the coordinate frame transformation between object and model.

By comparison the algorithms used during the model verification processes are designed less for speed and efficiency and more for rigorous verification. The constraints used are more complex relying, for the most part, on predictions made using the model to object coordinate frame transformation.

8.2 Numerical Algorithms Used During Part Selection

The numerical algorithms used during part selection can be divided into two sets. The first set concerns the initial estimates of the combined scale and stretch of the selected model part. The second set are algorithms to calculate the degree of fit between an object part and a deformed model part.

8.2.1 Estimation of the Combined Global Scale and Part Stretch Transform

During the model selection process, three object surfaces that are assumed to belong to the same object part are matched with three model surfaces from the same model part. The positions of the object surfaces in the world coordinate frame are known, while the positions of the model surfaces are defined according to the model part coordinate frame.

As defined in Chapter 4, the combined global scale and part stretch parameters S_{gx}, S_{gy} and S_{gz} represent the stretch transformation of an object part relative to that of the corresponding model part. Since, at this stage, the coordinate transformation between the model part coordinate frame and the world coordinate frame is unknown, the combined scale and stretch coefficients must be estimated independently of any rotation and translation between object and model. When estimating the deformation of an object, the criteria used are that any deformation should, as far as possible, be explained by a change in global scale. Any remaining deformation should be explained as far as possible by part stretching. Any remaining deformation is due to individual variation and should be minimised. If it is assumed that the individual variation in the positions of the object's surfaces is so small relative to the effects of scale and stretch that it can be disregarded, then Equation (6.10) holds true.

Given correspondences between three surfaces in the same part p, a set of three such equations are produced. Since each surface is from the same part they are subject to the same scale and stretch. By considering only the positive roots of the terms $S^2_{gx_p}$,

$S^2_{gy_p}$ and $S^2_{gz_p}$ the equations can be regarded as a set of simultaneous equations:

$$S^2_{gx_p}(x_1 - x_2)^2 + S^2_{gy_p}(y_1 - y_2)^2 + S^2_{gz_p}(z_1 - z_2)^2 = {D_1}^2 \tag{8.1}$$

$$S^2_{gx_p}(x_1 - x_3)^2 + S^2_{gy_p}(y_1 - y_3)^2 + S^2_{gz_p}(z_1 - z_3)^2 = {D_2}^2 \tag{8.2}$$

$$S^2_{gx_p}(x_2 - x_3)^2 + S^2_{gy_p}(y_2 - y_3)^2 + S^2_{gz_p}(z_2 - z_3)^2 = {D_3}^2 \tag{8.3}$$

To recap (x_1, y_1, z_1), (x_2, y_2, z_2) and (x_3, y_3, z_3) are the positions of the three model surfaces in the model coordinate frame and D_1, D_2 and D_3 are the distances between the corresponding object surfaces in the world coordinate frame.

Except for the cases covered in Section 6.2.2 where two or more of these equations are linearly dependent, they can be simply solved by LU decomposition [Press *et al*, 1988].

8.2.2 Determining the Degree of Fit for a Part Selection

Following the determination of the combined scale and stretch parameters, the selected model part can be deformed in the same way as the object part. To estimate how accurately the deformed model part represents the object part, differences between the orientations and curvatures of the component surfaces are used to calculate the degree of fit.

Estimating the Surface Normal of a Surface in a Stretched Model Part

A surface in the original model part has a normal vector $\mathbf{n}$. If the combined scale and stretch transformation given in Equation (4.4) is the transformation matrix S, then the normal vector to the surface in the deformed model part $\mathbf{n_d}$ is given by:

$$\mathbf{n_d} = \mathbf{S^{-1}n} \tag{8.4}$$

where $\mathbf{S^{-1}}$ is the inverse of the stretch transformation S.

Given a pair of surfaces from the object, the angle (from the dot product) between their surface normal vectors can be resolved in terms of their angle of slant and angle of tilt. The angles of slant and tilt can also be calculated for a pair of surfaces in the

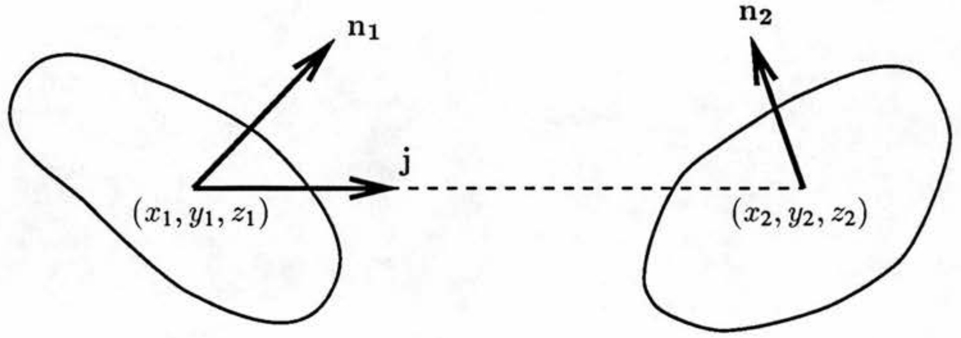


Figure 8–1: The Slant and Tilt Between the Normal Vectors of Two Surfaces

deformed model and these compared with those of the object to estimate the degree of fit.

In the method detailed below the slant and tilt are calculated for a pair of model surfaces. The same technique is used for a pair of object surfaces.

Estimation of the Slant and Tilt Between Two Surface Regions

Let $\mathbf{n_1}$ be the unit surface normal of the surface with centroid (x_1, y_1, z_1) and $\mathbf{n_2}$ be the unit surface normal of the surface with centroid (x_2, y_2, z_2) (see Figure 8–1). If the two surface centroids are joined by a line in coordinate frame space then, in the convention used here, the unit vector $\mathbf{j}$ points along this line from the centroid (x_1, y_1, z_1) towards the centroid (x_2, y_2, z_2).

The *angle of tilt* between the two surfaces is defined as the angle between vector $\mathbf{n_1}$ and the projection of vector $\mathbf{n_2}$ into the plane in which both $\mathbf{n_1}$ and $\mathbf{j}$ lie. Vector $\mathbf{i}$ (see Figure (8–2)) is the normal vector to the plane in which vectors $\mathbf{n_1}$ and $\mathbf{j}$ lie and

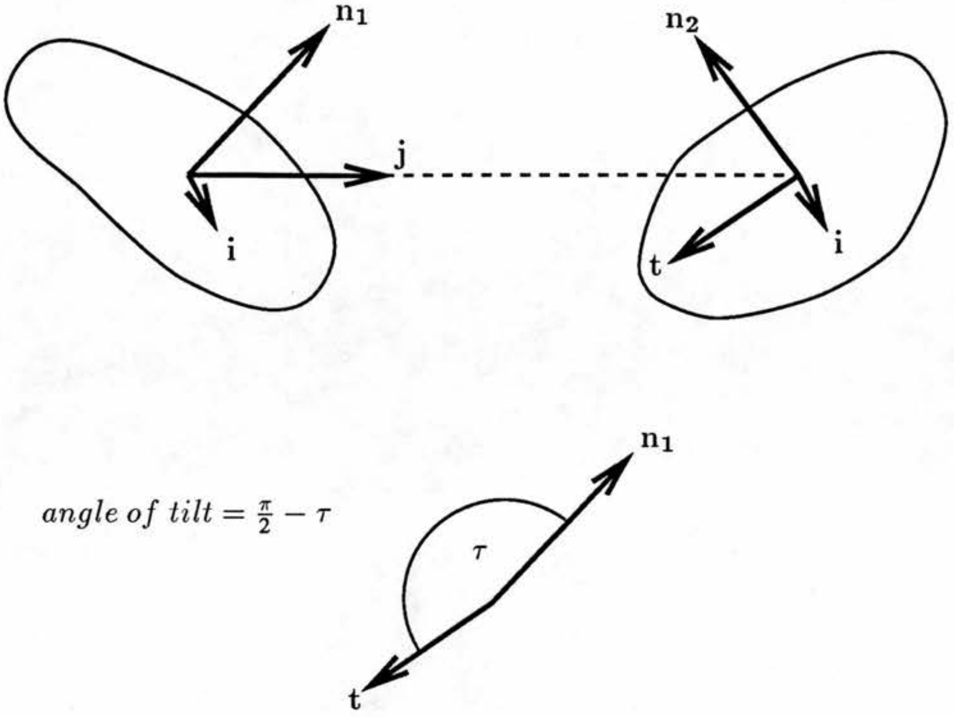


Figure 8–2: Tilt Between Two Model Surfaces

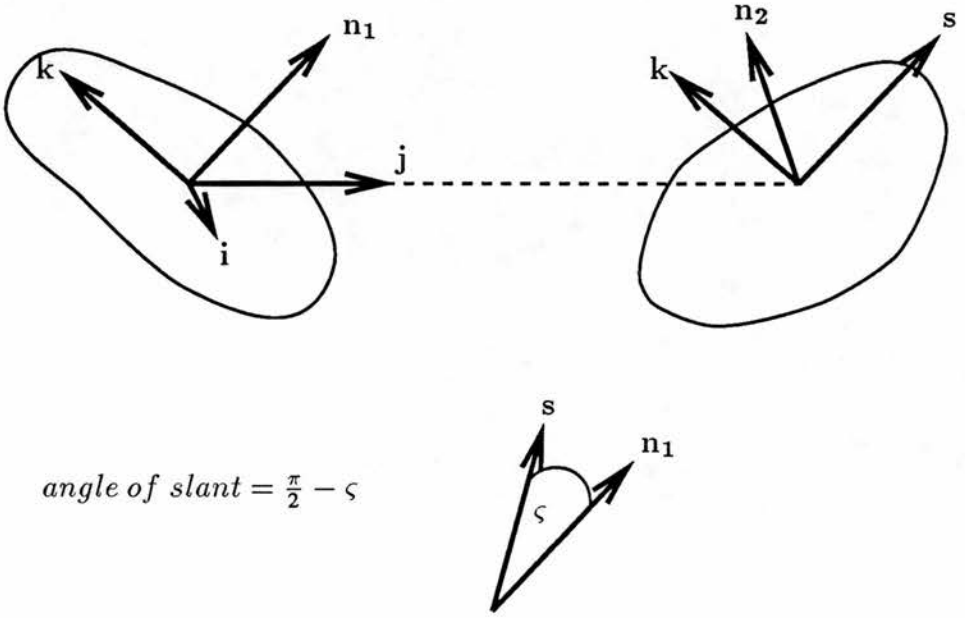


Figure 8–3: Slant Between Two Model Surfaces

may be obtained from the vector product:

$$\mathbf{i} = \mathbf{n_1} \times \mathbf{j} \tag{8.5}$$

Vector $\mathbf{t}$, the tilt vector, is the projection of $\mathbf{n_2}$ into the plane defined by $\mathbf{n_1}$ and $\mathbf{j}$ and is given by the vector product:

$$\mathbf{t} = \mathbf{n_2} \times \mathbf{i} \tag{8.6}$$

The vectors $\mathbf{n_1}$, $\mathbf{j}$ and the projection $\mathbf{t}$, all lie in the same plane and the angle τ between vectors $\mathbf{n_1}$ and $\mathbf{t}$ is obtained from the scalar product:

$$\tau = cos^{-1}\left(\frac{\mathbf{n_1}.\mathbf{t}}{|\mathbf{n_1}||\mathbf{t}|}\right) \tag{8.7}$$

The angle τ is in the range 0 to π radians. To yield a value between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$ this value is subtracted from $\frac{\pi}{2}$. Furthermore if the magnitudes of the vectors are normalised, the equation for the *angle of tilt* is given by:

$$angle\ of\ tilt = \frac{\pi}{2} - cos^{-1}(\mathbf{n_1}.\mathbf{t}) \tag{8.8}$$

The convention is that if $\mathbf{n_1}$ and $\mathbf{n_2}$ tilt "towards" each other, that is, in the absence of slant, $\mathbf{n_1}$ and $\mathbf{n_2}$ converge to a point above the surface, then the angle of tilt is said to be negative. This will generally occur where both surfaces lie in the vicinity of a concave (negative elliptical or negative cylindrical) region.

The *angle of slant* is defined as the angle between the vector $\mathbf{n_1}$ and the projection of the vector $\mathbf{n_2}$ onto the plane in which both $\mathbf{n_1}$ and $\mathbf{i}$ lie. Vector k, see Figure (8–3), is the normal vector to the plane in which vectors $\mathbf{n_1}$ and $\mathbf{i}$ lie and may be obtained from the vector product:

$$\mathbf{k} = \mathbf{n_1} \times \mathbf{i} \tag{8.9}$$

Vector s, the slant vector, is the projection of $\mathbf{n_2}$ into the plane defined by $\mathbf{n_1}$ and $\mathbf{i}$ and is given by the vector product:

$$\mathbf{s} = \mathbf{k} \times \mathbf{n_2} \tag{8.10}$$

The vectors $\mathbf{n_1}$, $\mathbf{i}$ and the projection vector s all lie in the same plane and the angle ς between the vectors $\mathbf{n_1}$ and s is obtained from the scalar product:

$$\varsigma = cos^{-1}\left(\frac{\mathbf{n_1}.\mathbf{s}}{|\mathbf{n_1}||\mathbf{s}|}\right) \tag{8.11}$$

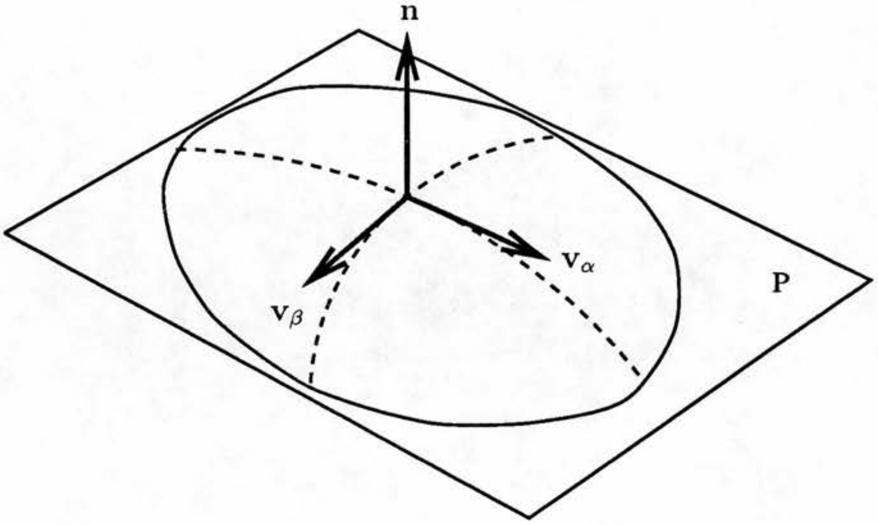


Figure 8–4: The Model Surface and its Tangential Plane

The angle ς is in the range 0 to π radians. To yield a value between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$ this value is subtracted from $\frac{\pi}{2}$. Furthermore if the magnitudes of the vectors are normalised, the equation for the *angle of slant* becomes:

$$angle\ of\ slant = \frac{\pi}{2} - cos^{-1}\left(\mathbf{n_1}.\mathbf{s}\right) \tag{8.12}$$

By convention, the angle of slant is positive when $\mathbf{n_2}$ appears slanted anti–clockwise to $\mathbf{n_1}$ when viewed from (x_1, y_1, z_1) in the direction of **j**.

Estimating the Curvature of a Surface in a Stretched Model Part

For a given model surface, the normal vector $\mathbf{n}$, is shared by a tangential planar surface P (see Figure (8–4)). The two principal curvatures of the surface κ_α and κ_β have orthogonal directions that can be represented by the two vectors $\mathbf{v}_\alpha$ and $\mathbf{v}_\beta$ that lie on the plane P. The vector product of these direction vectors is the surface normal vector $\mathbf{n}$.

When the model is deformed the directions of the vectors $\mathbf{n}$, $\mathbf{v}_\alpha$ and $\mathbf{v}_\beta$ in the original model are represented by $\mathbf{n}_d$, $\mathbf{v}_{d_\alpha}$ and $\mathbf{v}_{d\beta}$ in the deformed model. If the combined parameters of scale and stretch have already been determined then the vectors in the deformed model are given by Equation (8.4).

Once the direction vectors for the deformed model have been established then the effects of the scaling and stretching on the curvatures of the surface may be determined. The vectors all lie in the model coordinate frame and can be rewritten in terms of their components as:

$$\mathbf{n}_d = n_{dx}\mathbf{x} + n_{dy}\mathbf{y} + n_{dz}\mathbf{z} \tag{8.13}$$

$$\mathbf{v}_{d\alpha} = v_{d\alpha x}\mathbf{x} + v_{d\alpha y}\mathbf{y} + v_{d\alpha z}\mathbf{z} \tag{8.14}$$

$$\mathbf{v}_{d\beta} = v_{d\beta x}\mathbf{x} + v_{d\beta y}\mathbf{y} + v_{d\beta z}\mathbf{z} \tag{8.15}$$

where $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ are unit vectors along the axes of the model coordinate frame.

The principal curvatures $\kappa_{d\alpha}$ and $\kappa_{d\beta}$ of the deformed model surface are given by:

$$\kappa_{d\alpha} = \frac{\kappa_\alpha\sqrt{(v_{d\alpha x}S_{gx})^2 + (v_{d\alpha y}S_{gy})^2 + (v_{d\alpha z}S_{gz})^2}}{\sqrt{(n_{dx}S_{gx})^2 + (n_{dy}S_{gy})^2 + (n_{dz}S_{gz})^2}} \tag{8.16}$$

$$\kappa_{d\beta} = \frac{\kappa_\beta\sqrt{(v_{d\beta x}S_{gx})^2 + (v_{d\beta y}S_{gy})^2 + (v_{d\beta z}S_{gz})^2}}{\sqrt{(n_{dx}S_{gx})^2 + (n_{dy}S_{gy})^2 + (n_{dz}S_{gz})^2}} \tag{8.17}$$

where S_{gx}, S_{gy} and S_{gz} are the combined scale and stretch parameters determined earlier.

Equations (8.16) and (8.17) can be combined to calculate the product of the principal curvatures, the Gaussian curvature K_d, of the deformed model surface. By comparing the Gaussian curvature of the deformed model surface with that of the object surface, the degree of fit can be evaluated without the need to establish the direction of the principal curvatures in the object surface.

8.3 Numerical Algorithms Used During Superpart Selection

Unlike the part selection process the number of object to model correspondences used to make a superpart or model selection may vary. Obviously the more correspondences that are available to make the selection, the more accurately the object deformation can be calculated. This is reflected in the different methods used for different numbers of correspondences.

8.3.1 Estimation of the Combined Global Scale and Superpart Stretch Transform

Ideally, three object to model correspondences are available when selecting a model superpart or model. In this case the combined scale and stretch parameters $S_{gx_{sp}}$, $S_{gy_{sp}}$ and $S_{gz_{sp}}$ of the superpart sp can be determined as for part selection.

Where there are only two correspondences available, a different approach must be followed. The model to world coordinate transformation of each object part will have been determined during the part verification process. If the inverse transformation is used to transform the positions of the two object parts (X_1, Y_1, Z_1) and (X_2, Y_2, Z_2) to the corresponding positions in the model coordinate frame $(X_{TRD1}, Y_{TRD1}, Z_{TRD1})$ and $(X_{TRD2}, Y_{TRD2}, Z_{TRD2})$, then the combined scale and stretch factors can be calculated directly from their positions in the model superpart coordinate frame relative to the positions of the corresponding model parts (x_1, y_1, z_1) and (x_2, y_2, z_2) in the same frame:

$$S_{gx_{sp}} = \frac{\frac{X_{TRD1}}{x_1} + \frac{X_{TRD2}}{x_2}}{2} \tag{8.18}$$

$$S_{gy_{sp}} = \frac{\frac{Y_{TRD1}}{y_1} + \frac{Y_{TRD2}}{y_2}}{2} \tag{8.19}$$

$$S_{gz_{sp}} = \frac{\frac{Z_{TRD1}}{z_1} + \frac{Z_{TRD2}}{z_2}}{2} \tag{8.20}$$

An alternative technique would be to use the x, y and z components of the distances between parts in both pairs as a measure of combined scale and stretch, but this would give an estimate based on only one value, whereas the approach used here is based on the average of two values.

Unlike the three correspondence case, this approach is dependent on the accuracy to which the position and orientation of the two component parts has been calculated. Any inaccuracy or error here would be propagated into the estimate of the combined scale and stretch of the superpart.

Where there is only one correspondence available, the combined scale and stretch parameters of the superpart are assumed to be those of the component part suitably transformed into the coordinate frame of the superpart by the transformation specified in the model.

8.3.2 Estimation of the Optimal Global Scale Transform

As far as possible, a deformation is to be explained in terms of a change in scale in preference to a stretch or a variation. Given this criteria the residual stretching exhibited by the parts, superpart and object should average out to unity. If there are n object to model part correspondences in the selection then the global scale is estimated by finding the geometric average of the combined scale and stretch parameters of each of the n object parts and the combined scale and stretch, if available, of the superpart itself:

$$S_g = \sqrt[3(n+1)]{S_{gx_1} S_{gy_1} S_{gz_1} \ldots\ldots\ldots S_{gz_{n+1}}} \qquad (8.21)$$

As superparts further up in the model hierarchy are selected, the global scale is continually updated, averaging the combined scale and stretch of all the component parts and superparts.

8.4 Numerical Algorithms Used During Part Verification

The intention of the part verification process is to verify the existence of an object in the image by finding correspondences for all the features modelled in the part. The algorithms here are used to recalculate the combined scale and stretch, the rotation and the scale of the model part as these new correspondences are found.

8.4.1 Estimation of the Optimal Model Part Transformation

During part verification, it is necessary to establish the coordinate frame transformation between the model part coordinate frame and the world coordinate frame. The full coordinate frame transformation between a point (x_i, y_i, z_i) in a model part p and the corresponding point in the image (X_i, Y_i, Z_i) is given by:

$$\begin{pmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{pmatrix} = \begin{pmatrix} & & & t_{xpw} \\ & R_{pw}(r,s,t) & & t_{y_{pw}} \\ & & & t_{zpw} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} S_{gx_p}V_{xi} & 0 & 0 & 0 \\ 0 & S_{gy_p}V_{y_i} & 0 & 0 \\ 0 & 0 & S_{gz_p}V_{zi} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} \tag{8.22}$$

where $R_{pw}(r,s,t)$ and t_{xpw}, $t_{y_{pw}}$ and t_{zpw} are the coordinate frame rotation and translation respectively, S_{gx_p}, S_{gy_p} and S_{gz_p} are the combined scale and stretch parameters of the part p and V_{xi}, V_{y_i} and V_{zi} are the individual variations in position for the point.

It is often the case that there are more model to image correspondences available than the minimum necessary to uniquely determine the transformation, so the *optimal* transformation must be obtained for all the object to model pairings available.

Estimation of the Optimal Combined Global Scale and Part Stretch Transform

Where there are only three object to model correspondences, the combined effects of global scale and stretch parameters of the object part S_{gx_p}, S_{gy_p} and S_{gz_p} are obtained as above by solving Equations (8.1), (8.2) and (8.3).

Where there are four or more object to model correspondences, in the absence of any linearly dependent expressions, the equation set becomes over–determined. Due to the effects of individual variation there is unlikely to be an exact solution. Any deformation should be explained as far as possible by a scale change then by a part stretch. Any residual due to individual variation should therefore be minimised.

If $\mathbf{A}$ is the $3 \times n$ matrix containing the coefficients of the equation set:

$$\begin{pmatrix} (x_1 - x_2)^2 & (y_1 - y_2)^2 & (z_1 - z_2)^2 \\ (x_1 - x_3)^2 & (y_1 - y_3)^2 & (z_1 - z_3)^2 \\ .... & .... & .... \\ (x_i - x_j)^2 & (y_i - y_j)^2 & (z_i - z_j)^2 \end{pmatrix}$$

where $j > i$.

b is the $1 \times n$ vector containing the results of the equations:

$$\begin{pmatrix} D_1^2 \\ D_2^2 \\ .... \\ D_n^2 \end{pmatrix}$$

and x is the 1×3 vector containing the scale and stretch factors:

$$\begin{pmatrix} S_{gx_p}^2 \\ S_{gy_p}^2 \\ S_{gz_p}^2 \end{pmatrix}$$

then the best fit solution is the vector x that minimises the expression:

$$\mid \mathbf{A}.\mathrm{x} - \mathbf{b} \mid \tag{8.23}$$

Equation (8.23) will only minimise to zero in the absence of any individual variation. Usually this is not the case but the best fit vector x will be that which minimises the variation.

The best fit value of x is obtained by the use of singular value decomposition [Press *et al*, 1988].

The position of a point (x_i, y_i, z_i) in the original model is mapped to a point (x_{di}, y_{di}, z_{di}) in the deformed model by the transformation given in Equation (4.4):

$$\begin{pmatrix} x_{di} \\ y_{di} \\ z_{di} \\ 1 \end{pmatrix} = \begin{pmatrix} S_{gx_p} & 0 & 0 & 0 \\ 0 & S_{gy_p} & 0 & 0 \\ 0 & 0 & S_{gz_p} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} \tag{8.24}$$

Estimation of the Best Rotation

Although the model part is deformable, the deformation parameters of the object part are known and the model part has been deformed accordingly. The transformation between the deformed model part and the object part can therefore be regarded as a *rigid* transformation. Where the deformed model part is an *exact* representation of the object part, correspondences between the sets of direction vectors joining the positions of the surfaces are sufficient to establish an exact rotation between the coordinate frame of the model part and that of the world $R_{pw}(r, s, t)$. Here, however, the deformed model part is usually not an exact representation of the object due to the effects of variation in the object. Where there are more than three model to object correspondences there will be more than two sets of vectors available. As before, the rotation must be determined for the model part such that any residual difference between model part and object is minimised.

The best rotation is determined by the least–squares fit method developed by Faugeras [Faugeras & Hebert, 1983]

The position of a point (x_{di}, y_{di}, z_{di}) in the deformed model is mapped to a point $(x_{dri}, y_{dri}, z_{dri})$ in the deformed and rotated model by the transformation given in

Equation (4.5):

$$\begin{pmatrix} x_{dri} \\ y_{dri} \\ z_{dri} \\ 1 \end{pmatrix} = \begin{pmatrix} & & & 0 \\ & R_{pw}(r,s,t) & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{di} \\ y_{di} \\ z_{di} \\ 1 \end{pmatrix} \tag{8.25}$$

Estimation of the Optimal Translation

With the combined scale and stretch and rotation parameters known, it only remains to find the translation $(t_{xpw}, t_{y_{pw}}, t_{zpw})$ between the model part coordinate frame and the world. Since all surfaces in the model part are subject to the same translation, the best translation is simply the average translation of all the surfaces:

$$t_{xpw} = \frac{1}{n}\sum_{i=1}^{n}(X_i - x_{dri}) \tag{8.26}$$

$$t_{y_{pw}} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - y_{dri}) \tag{8.27}$$

$$t_{zpw} = \frac{1}{n}\sum_{i=1}^{n}(Z_i - z_{dri}) \tag{8.28}$$

where n is the number of surface correspondences and (X_i, Y_i, Z_i) are the positions of the object's surfaces in the world coordinate frame.

The position of a point $(x_{dri}, y_{dri}, z_{dri})$ in the deformed and rotated model is mapped to a point $(x_{drti}, y_{drti}, z_{drti})$ in the deformed, rotated and translated model by the transformation given in Equation (4.5):

$$\begin{pmatrix} x_{drti} \\ y_{drti} \\ z_{drti} \\ 1 \end{pmatrix} = \begin{pmatrix} & & & t_{xpw} \\ & R_{pw}(r,s,t) & & t_{y_{pw}} \\ & & & t_{zpw} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{dri} \\ y_{dri} \\ z_{dri} \\ 1 \end{pmatrix} \tag{8.29}$$

Estimation of the Individual Variation

The combined scale and stretch parameters and the coordinate frame transformation between the model part and the object are optimal in that they minimise the effects of individual variation. However, the variations in the features of a surface will not usually be zero. With the full coordinate transformation known, the individual variation for the i^{th} surface is merely the difference between that of the object and that of the deformed model. For the effects of variation to be of use to the verification process, they must be calculated in the model coordinate frame so that they can be applied to identical model features. The variation parameters V_{xi}, V_{y_i} and V_{zi} for the position of a surface i are obtained by using the *inverse* of the coordinate transformation to transform the object surface into the model coordinate frame. If the position of the surface in the model coordinate frame is (x_i, y_i, z_i) and the position of the transformed object surface is $(X_{TRDi}, Y_{TRDi}, Z_{TRDi})$ then the variation in the position of that surface is given by:

$$V_{xi} = \frac{X_{TRDi}}{x_i} \tag{8.30}$$

$$V_{y_i} = \frac{Y_{TRDi}}{y_i} \tag{8.31}$$

$$V_{zi} = \frac{Z_{TRDi}}{z_i} \tag{8.32}$$

The values calculated in this manner for the individual variation are multiplicative factors rather than, in this case, a simple translation vector.

8.5 Numerical Algorithms Used During Superpart (or Model) Verification

The function of the superpart verification process is very similar to the part verification process and the algorithms described here reflect this. However, the superpart verification process generally deals with combinations that contain fewer parts but about which more information is known. The algorithms described here reflect this.

8.5.1 Estimation of the Optimal Model Superpart (or Model) Transformation

As with the part verification process during the superpart (or model) verification process, it is necessary to establish the best coordinate frame transformation between the superpart and the object. The part verification and superpart verification processes are almost identical but unlike the part verification process the number of model to object correspondences can vary. The method by which an optimal transform is determined will depend on the number of correspondences available.

Estimation of the Optimal Combined Global Scale and Superpart Stretch Transform

The combined global scale and stretch of a superpart is initially estimated during the superpart selection process. If further correspondences are determined then the parameters are recalculated using the appropriate method. Where two correspondences are available, Equations (8.18), (8.19) and (8.20) are used. With three or more correspondences then the scale and stretch parameters of the superpart can be determined (see Section 8.4.1), as described in the part verification process, independently of its component parts.

As new correspondences are found the global scale of the object is recalculated using Equation (8.21).

Estimation of the Optimal Rotation

If there are only one or two correspondences in the initial selection, the coordinate frame transformation between the superpart and the world must be estimated from the transformation between the component parts and the world and the transformation between part and superpart as represented in the model. Where there are three or more correspondences then the optimal rotation is estimated as for part verification (see Section 8.4.1).

Estimation of the Optimal Translation

When calculating the translation of the superpart, Equations (8.26), (8.27) and (8.28) can be applied for any number of correspondences.

Estimation of the Individual Variation

Before the individual variations of the parts within a superpart can be calculated four correspondences are required. Variation is then calculated as for part verification (see Section 8.4.1).

8.5.2 Estimation of the Stretch of the Component Parts

So far the deformation of an object has been described in terms of the combined effects of global scale and all levels of part stretch. During model verification it is necessary to resolve these effects into the global scale and the stretch of each superpart and part.

The global scale S_g of the object is estimated as in Equation (8.21) for each part, superpart and the object itself. With the global scale known, the combined scale and stretch parameters of the object $S_{gx_{object}}$, $S_{gy_{object}}$ and $S_{gz_{object}}$ can be resolved by simply dividing through by S_g to yield the object stretch parameters:

$$S_{x object} = \frac{S_{gx_{object}}}{S_g} \tag{8.33}$$

$$S_{y_{object}} = \frac{S_{gy_{object}}}{S_g} \tag{8.34}$$

$$S_{z_{object}} = \frac{S_{gz_{object}}}{S_g} \tag{8.35}$$

These stretch parameters are transformed into the coordinate frame of each superpart or part at the next level down in the model hierarchy. The stretch parameters of a superpart sp are found by simply dividing by the global scale and the transformed object stretch parameters – *the combined scale and stretch parameters of the object*:

$$S_{x_{sp}} = \frac{S_{gx_{sp}}}{S_{gx_{transobject}}} \tag{8.36}$$

$$S_{y_{sp}} = \frac{S_{gy_{sp}}}{S_{gy_{transobject}}} \tag{8.37}$$

$$S_{z_{sp}} = \frac{S_{gz_{sp}}}{S_{gz_{transobject}}} \tag{8.38}$$

This process is repeated for each level in the hierarchy, the stretch parameters of any part being obtained by dividing the combined scale and stretch parameters by the transformed combined scale and stretch parameters from the level above.

8.6 Conclusion

In this chapter the numerical methods used in the model selection and model verification processes have been described. The methods used during model selection rely on relatively simple constraints based on information available locally for fast and efficient matching. Unfortunately the rather simple criteria used in some of the constraints, such as always assuming that the individual variation is negligible, can lead to incorrect selections or more often, selections with inaccurate parameters. The number of selections is normally very large and since each object is usually selected by several selections, the occasional incorrect selection is a small price to pay for the increased efficiency.

The methods used during model verification emphasise rigorous identification at the cost of speed and efficiency. The constraints used here rely on information global to the part or superpart such as its position and orientation. The continual recalculation of the object's parameters means that as new correspondences are found the constraints can be tightened. There are less model verifications than selections so that the mis–identification of an object during verification would have far greater consequences.

Chapter 9

Conclusions

9.1 Summary

This thesis has considered the problem of three dimensional object recognition using parameterised, or deformable, three dimensional models to represent classes of objects rather than individual examples. A representation suitable for modelling classes of objects has been proposed and the techniques by which the identity of an object, its position and its parameters can be determined has been described.

9.1.1 Representing Classes of Objects

The objects in the image are described in terms of surface regions or patches. Surfaces are also used as the basic modelling primitives in the construction of the models. This aids the recognition process by allowing direct matching between the object data and that of the models.

The models used to represent classes of objects have a hierarchical structure. At the lowest levels of the models, model features are represented by surfaces. One level up in the hierarchy, configurations of surfaces form model parts. Further still up the hierarchy configurations of model parts and larger model surfaces form model superparts. At the highest level the model itself is defined by a configuration of superparts, parts and surfaces. The positions of the surfaces within a part are defined within the coordinate frame of the part rather than that of the model. The position of the part is defined in terms of the coordinate frame of the superpart of which it is a component. This occurs at each level of the model hierarchy with the position of a part or superpart being defined in terms of the coordinate frame of the level immediately above.

The model parameters defined enable the deformation of the corresponding object relative to the original model to be represented. By setting the values of the model parameters to be the same as those of a given object, the model may be deformed to match the object. Three sets of deformation parameters are defined:

- The global scale – regulates the overall scale of an object.
- The part stretch – represents an elongation or contraction along up to three orthogonal axes in each part and superpart of the object and in the object itself.
- Individual variation – represents small perturbations in the properties of individual object features not accounted for by either scale or stretch.

The hierarchical structure of the model is reflected in the structure of the deformation parameters. At the highest level of the model hierarchy the global scale parameter controls the overall size of the whole model. For each model part, superpart and the model itself there is a set of part stretch parameters. The stretch of a part does *not* represent the relative stretch between the model part and the corresponding object part but the stretch of the part relative to the stretch of the superpart or model at the next level up in the hierarchy. Individual variation is normally present at all levels of the hierarchy but affects only the properties of the individual features at that level.

9.1.2 Object Recognition

Given an image containing an unknown number of objects and a database containing a large number of models it is inefficient, and often impractical, to attempt to match each object in turn with each model in the database. A more promising approach, and the approach taken here, is to quickly and efficiently identify distinctive features in the image and use these to select a reduced set of model parts for more rigorous matching.

However, because the objects in the image may be deformed relative to the model that represents that class of object, the appearance of its features, in this case represented by surfaces, may be very different from the corresponding surfaces in the model. Without knowing the deformation of the objects in the image in advance, a single object surface in isolation yields little evidence as to its identity and the object of which it forms part.

Since a single object surface is no longer distinctive enough to give a clue as to its identity then combinations of three object surfaces are used. It is assumed that each surface in a combination belongs to the same object part and therefore they are only matched with different permutations of surfaces from each model part. An interpretation tree is used to achieve this; one interpretation tree being generated each time a combination of object surfaces is matched with the surfaces of a model part. The depth of the interpretation tree is limited by the number of object surfaces in the combination and not the number of object surfaces in the image. The width of the interpretation tree is limited by the number of surfaces in the model part.

Since each object surface in the combination is assumed to belong to the same object part, then they are assumed to be subject to the same deformation in scale and part stretch. As the interpretation trees are generated and object surfaces matched with model surfaces, the scale and stretch exhibited by the object surfaces relative to those of the model are determined.

The effects of each type of deformation are interrelated and can appear the same so in separating the effects of each, two basic criterion are used:

- Firstly, any deformation should, as far as possible, be explained by a change in the global scale.
- Secondly, any remaining deformation should be explained by part stretching so that any individual variation is minimised.

The techniques used enable the deformation parameters to be established independently of the position and orientation of the object part in the image using only "local" data. This increases the speed with which interpretations can be generated by avoiding the need for time consuming geometric transformations.

Each branch of each interpretation tree represents a possible interpretation of a combination of object surfaces. Unfortunately, not only are the majority of these interpretations incorrect but the number generated is usually so large as to make matching impractical. To overcome this, a set of constraints have been developed to identify incorrect object to model matches:

1. The shape of an object surface, defined by the signs of its Gaussian and mean curvatures, must be a plausible match with that of the corresponding model surface.

2. The combined scale and stretch of an object, calculated along the line joining two object surfaces, must be within the maximum and minimum limits of combined scale and stretch specified for that model part.

3. The combined scale and stretch exhibited by three object surfaces along the three axes of the model part coordinate frame must fall within the maximum and minimum limits imposed by that model part.

4. The sign of the triple scalar product of the direction vectors of the three object surfaces and their surface normals must be the same as for those of the model.

Where a match, or matches, are found that are inconsistent with any of these constraints the interpretations of which they form part are removed, or "pruned", from the interpretation tree. The remaining interpretations are not necessarily correct but this reduced set provides the basis for more rigorous matching.

Having selected a subset of model parts, the next stage in the process is to verify that each of the model parts selected correctly explains the data in the image. This is achieved by taking each selected model part in turn and searching the image for object surfaces to match any unmatched model surfaces. Due to self–occlusion, not all surfaces represented by the model may be visible in the image. To predict the appearance of an object and where in the image an object surface is likely to be found, the transformation that maps the model part to that of the object is first determined. Since the original model part may not be an exact representation of the object, finding an exact transformation between the two is not possible. However, by using the deformation parameters determined earlier to deform the model part in the same way as the object part, an exact "rigid" transformation between deformed model and object can be determined and used in the prediction of further object features.

Any unmatched features are searched for one by one on a "generate and test" basis. A further search tree is used to represent possible matches between model and object. The structure of this tree is the opposite to that of the interpretation tree in that the breadth of the tree is dictated by the number of object surfaces in the image and its depth by the number of surfaces in the model part. Each level in the tree represents the set of possible matches for a given model surface. The size of this search tree is also potentially very large. Since the transformation between deformed model part and object part is known, the position and appearance of any unmatched surface represented by the model can be predicted. This provides further constraints, this time based on data "global" to the part, as to which of the object surfaces are suitable matches, thus enabling this search tree to be pruned too.

As new model to object correspondences are found then the estimates of the coordinate frame transformation and the deformation parameters are updated. With a correspondence of more than three surfaces there is unlikely to be an exact transformation, so one that is optimal for the part as a whole must be determined. This is achieved in a series of steps:

1. The scale and stretch of the part is determined using a least–squares fit so that the effects of individual variation associated with the distances between each surface are minimised.

2. The model part is deformed and the rotation that maps the surfaces of the model to those of the object is determined such that any remaining variation in surface orientation is minimised.

3. The deformed model part is rotated and the translation determined that minimises any remaining difference in position between the surfaces of the model and those of the object.

If each model surface represented in the model part is either predicted as not being visible in the image or the corresponding object surface is found then the presence of the object part is regarded as verified. If a model feature that is predicted as visible in the image is not found then the original interpretation is regarded as false and deleted.

Once object parts have been identified in the image, combinations of them are used to select the model superparts from the database for further matching. Since the identity of these object parts is known, it is not necessary to match them with the parts of each model superpart. However, several parts of the same type of superpart may be visible. An interpretation tree is used to match a combination of object parts with permutations of parts from a model superpart to determine if the parts are consistent with belonging to the same superpart. The scale and stretch of the selected superpart is determined and constraints similar to those described earlier are used to identify inconsistent interpretations.

Verification of any selected superparts proceeds by checking that each component superpart, part and surface is visible in the image or that its absence can be explained. This process of selection followed by verification is repeated until there are no more selections left to verify.

9.2 Summary of the Results

The techniques described in this thesis have been tested on images containing a range of objects, both synthetic and real, and shown to correctly and efficiently identify the objects and give good estimates of their deformation parameters.

9.2.1 The Reliability of Recognition

The next two sections discuss the reliability of the model selection and model verification processes. The reasons why inaccurate identification were made are discussed.

The Reliability of Model Selection

The vast majority of object surface combinations generated had no corresponding permutation of model surfaces. The exact proportion varied with the type and number of objects in the image but, in the images used here, was between 75% and 97% of the combinations generated. It was therefore the case that interpretations of object surfaces would be expected where in fact there was no valid interpretation. The number of incorrect interpretations also varied; for the images used here it was between 0% and 79% of the combinations for which interpretations were found. However, what was more important was that those combinations that did have valid interpretations should be correctly identified.

The accuracy with which an object part was identified during the model selection process proved satisfactory. At least three surfaces of the part had to be visible in the image and correct interpretation was achieved provided they met the specified constraints.

Failure to identify, or incorrect identification of, an object part was extremely rare, less than 2% if it occurred at all, but where it did occur it was usually for one of the following reasons:

1. The shape of an object surface did not match that of the corresponding model surface. This was usually due to either variation in the curvature of an object surface changing its shape or noise in the original range data causing the curvature to be calculated incorrectly. The latter occurred most often with very small surface regions.

2. The combined scale and stretch of the object part fell outside the limits imposed by the corresponding model part. This usually occurred where the distance between two model surfaces along at least one axis in the model coordinate frame was small. Any errors in the corresponding object distance generated large changes in the scale and stretch parameters causing them to fall outside the allowed limits.

The accuracy with which superparts were selected depended on the accuracy of the part verification process. If a part was correctly verified then the correct superpart would be selected. Inaccurate estimates of scale and stretch occurred where two or three parts were incorrectly considered to belong to the same superpart. This occurred only rarely because of the relatively small number of object parts verified. Where it did occur, verification would then proceed using the same selection based on fewer object parts until, if necessary, selections based on only one object part would be used.

The Reliability of Model Verification

The accuracy with which the model verification process was able to verify the identity of object parts proved less than satisfactory. This was due mostly to the original segmentation of the range data into surfaces with no matching surface in the corresponding model part. Given a correct and accurate selection, the verification process was able to accurately predict the position and shape of any unmatched model surfaces. Unfortunately this was often not as the surface had been segmented in the image. The tests with the synthetic images and the set of screwdrivers shows that the verification process works well where the surface segmentation is as expected.

The problem is due mostly to the rather vague surface descriptions used in the modelling system. Possible solutions to this problem are discussed in later sections.

9.2.2 Estimating the Parameters of Deformation

Here the accuracy with which the deformation parameters were estimated in both the model selection and model verification processes is discussed. The conditions that give rise to inaccurate estimates are also discussed.

Estimation of Deformation During Model Selection

During the model selection process, the basic assumption is made that the positions of the three object surfaces used to make the selection show *no* individual variation. Except in the case of synthetically generated images this assumption is unlikely to be true. However, if the individual variation is small, then its effect on the scale and stretch parameters should also be small and the estimated values should at least provide a good starting point for the verification process. Individual variation can occasionally have a large effect on the estimates of combined scale and stretch where the corresponding distance in the model is small. A small change in the position of an object surface can cause a large change in its estimated scale and stretch. This can be avoided to some degree by careful construction of the model. Alternatively if there are many surfaces in the object part, a combination of different surfaces should make the same selection but with more accurate parameters.

The synthetic image used to test the effects of individual variation showed that correct selections with reasonable estimates of scale and stretch could be made in the presence of positional variation and that combinations of other surfaces exhibiting less (or no) variation could be identified and ranked in preference for the verification process.

The process worked well with real images, though if there were significant errors in the segmentation then the estimates of scale and stretch were poor.

The estimation of the global scale proved robust, though obviously the results were better the more sets of independent scale and stretch data that were available.

Estimation of Deformation During Model Verification

To verify a model selection, the verification process is reliant on a reasonable estimate of the combined scale and stretch. The estimate does not have to be accurate, merely sufficient to predict the position of the next unmatched part with sufficient accuracy for a correct correspondence to be achieved. After that the accuracy of the estimates of the scale, stretch, rotation and translation of the object would increase as further correspondences were found.

The images that exhibited stretching and individual variation showed that even a selection made with poor estimates of scale and stretch could be verified. When the remaining correspondences were found then the recalculated estimates would be the same as those obtained from verifying selections made with better original estimates. The degree to which this was achieved depended on the number of correspondences available. In the "worst case" screwdriver images where just one part consisting of three surfaces was available then the estimates could not be recalculated and both these and the resulting rotation and translation were susceptible to errors and variation. In the synthetic image, the larger numbers of correspondences available ensured that the parameters could be calculated to greater accuracy.

The final estimates of the resolved global scale and the stretch of the object parts were also dependent on the number of correspondences made.

9.2.3 The Efficiency of Recognition

In the section the efficiency with which combinations of object features are interpreted during the model selection process and the resulting set of interpretations verified is discussed. The maximum number of verified matches is shown to compare favourably with previous research.

The Efficiency of Model Selection

Previous work has shown that the traditional interpretation tree search is at its most efficient where all the features in the image belong to the same object. Where this is the case, the use of constraints can reduce the number of interpretations to a quadratic function of the number of model features. If, however, the image contains spurious features or features from different objects then the number of interpretations becomes an exponential function of the number of object features visible in the image. Careful use of heuristically determined thresholds can reduce the number of interpretations to a quartic function but only if the amount of spurious features is limited.

In the approach described here, interpretations are found not for the entire set of features in the image but only for each combination of three. The number of interpretation trees generated is much greater but the size of each tree is greatly reduced. The width of the tree is limited by the number of model features and its depth is limited to three by the number of image features for which an interpretation is required. Because the number of interpretations produced by each interpretation tree is fixed, the total number of interpretations produced is a function of the number of combinations of three object features. For an image containing I object features corresponding to a model of m features, the total number of interpretations produced is given by:

$$number\ of\ interpretations = \frac{I!}{(I-3)!3!}\frac{m!}{(m-3)!} \tag{9.1}$$

If I and m are both much larger than three, then the number of interpretations is very roughly approximated by a cubic function of their product.

The maximum number of interpretations generated is slightly more than an unconstrained standard interpretation tree search and suffers the disadvantage that only three object features are included in any interpretation, whereas the standard interpretation tree search will produce an interpretation for them all.

The advantage of the model selection approach occurs where an image contains features from different objects. The number of interpretations produced by even a constrained interpretation grows exponentially, whereas the number of interpretations generated by the model selection process is given by:

$$number\ of\ interpretations = \frac{I!}{(I-3)!3!}\sum_{i=1}^{p}\frac{m_i!}{(m_i-3)!} \tag{9.2}$$

where there are p models each with m_i features. Since m_i and p are small, the number of interpretations is approximated by a cubic function of I.

This is the *maximum* number of interpretations that can be generated and this number is reduced further by the application of constraints often returning only the correct interpretations.

The Efficiency of Model Verification

The model verification process is a simple tree search as used in numerous examples of previous work. This process benefits from the three correspondences initially made by the model selection process. These not only reduce the number of correspondences still remaining to be found, but also enable the transformation between deformed model part and object part to be found. With the coordinate transformation between the two known then the search for any remaining unmatched model features in the image can be heavily constrained.

In the *absence* of any constraints, the maximum number of verified matches is given by:

$$number\ of\ verified\, matches\ = \frac{I!}{(I-3)!3!}\sum_{i=1}^{p}\frac{m_i!}{(m_i-3)!}I^{(m_i-3)} \tag{9.3}$$

If *I* is much greater than three, the function becomes:

$$number\ of\ verified\, matches \approx \sum_{i=1}^{p}\frac{m_i!}{(m_i-3)!}I^{m_i} \tag{9.4}$$

The maximum number of possible verified interpretations without the application of any constraints is now approximated by an exponential function of the number of model features. Since, for a given image, the number of model features is a constant, the maximum number of verified matches is approximated by a polynomial function of the number of object features in the image. Fortunately, because this function increases only linearly with the number of models p then the search space can be made smaller by reducing the number of features m_i in each model and effectively increasing the number of models by the division of models into more numerous model parts.

9.3 Comparison with Previous Work

The three main aspects of the work described in this thesis, that of the representation of deformable objects, the selection of models from the model database and the estimation of the model parameters, and the verification of those selections and the recalculation of the model parameters are compared with previous research.

9.3.1 Object Representation

The representation of classes of object by parameterised models has been covered in a great deal of previous work [Roberts, 1965], [Brooks, 1983], [Fisher, 1985], [Ayache & Faugeras, 1986], [Huttenlocher & Ullman, 1987], [Murray, 1987b] and [Grimson, 1987 Of these, the last most closely resembles the work described here. In Grimson's work objects could show changes in scale, stretch along a single axis or rotation between two parts, though an object could not show these features simultaneously.

The global scale defined in this work regulates the scale of an object as a whole such as that defined in [Roberts, 1965], [Fisher, 1985], [Huttenlocher & Ullman, 1987] and [Murray, 1987b] but *not* the size of the object as represented in the image (essentially a camera calibration parameter) as defined in [Ayache & Faugeras, 1986]. The scale defined by Grimson regulated not the scale of an object but the scale of an object part. The scale was allowed to vary between object parts though it had to be "roughly the same" for a consistent match. In the work defined here, the scale is constant throughout the object and all its parts.

The stretch of an object part, while similar to that of Grimson's work, has been extended to object parts of three dimensions. Furthermore, the stretch of an object and its parts is not defined in absolute terms but instead is defined in terms of the hierarchical structure of the model. Model parameters with a range, or structure, have been used in previous work [Brooks, 1983] and [Fisher, 1985].

The last type of object parameter, that of individual variation, has no equivalent in previous work, though it is similar to the "rubber mask" technique [Fischler & Elschlager, 1973] and related to the error spaces that the properties of object features were allowed to occupy in previous recognition systems.

9.3.2 Model Selection

The concept of attempting to identify which objects are present in an image before subjecting the corresponding models to a rigorous geometric test is not a new one. This has been approached in a variety of ways [Bolles & Cain, 1982], [Gaston & Lozano-Pérez, 1984], [Grimson & Lozano-Pérez, 1984], [Fisher, 1986], [Ayache & Faugeras, 1986], [Murray, 1987b], [Huttenlocher & Ullman, 1987] and [Pollard *et al*, 1987]. The model selection process described here is a combination of the interpretation tree search [Gaston & Lozano-Pérez, 1984] and the local feature focus method [Bolles & Cain, 1982]. There are similarities with work done on the TINA project [Pollard *et al*, 1987] with the work described here, in that object features are used to drive the generation and limit the size of the interpretation trees. However, the way in which this was achieved was substantially different and the crossover to verification less obvious. Furthermore, no previous attempt was made to approach the problem of recognition using parameterised models.

9.3.3 Model Verification

Where an object is thought to be present in the image, this is normally verified by attempting to estimate a consistent rigid transformation that maps the features of the model to the corresponding features of the object. This is the technique used in most pieces of work that approach this problem [Faugeras & Hebert, 1983], [Gaston & Lozano-Pérez, 1984], [Grimson & Lozano-Pérez, 1984], [Goad, 1985], [Murray, 1987b [Lozano-Pérez *et al*, 1987], [Huttenlocher & Ullman, 1987] and [Pollard *et al*, 1987]. This approach is followed in the work described here. Although the objects used in this work are allowed to deform, the deformation parameters are calculated in advance enabling the model to be deformed in advance and the *rigid* transformation between deformed model and object to be determined using standard techniques.

9.4 Limitations and Possible Extensions

The limitations of the representation chosen to represent deformable objects and the model selection and model verification processes are discussed in detail and suitable modifications and extensions proposed.

9.4.1 Limitations and Extensions in Object Representation

Three types of deformation parameters are defined global scale, part stretch and individual variation. The scale of an object and the stretch of its parts are determined using several object features and although affected by noise or positional errors, these are minimised by the least–squares methods used to derive them. The individual variation on the other hand is determined purely on the properties of one particular feature. Any noise or errors associated with this feature are incorrectly regarded as individual variations. Fortunately, individual variation is not normally used as part of the recognition process and so this limitation has little or no effect on the accuracy of the process as a whole.

Other forms of deformation such as shearing, tapering and bending were not covered in this work as these were thought to be less common in object classes than scaling, stretching and variation. Of the three, bending would probably be the easiest to add. The surface normals are largely unused in the model selection process and these could be used to give an estimate of any part bending. As with part stretching, part bending could be represented hierarchically with the bend of a part defined in terms of the bend of the superpart of which it was a component.

The work most closely related to this [Grimson, 1987] also dealt with two dimensional object parts that rotated, though not those that simultaneously showed scale or stretch. The recognition of three dimensional objects with freely rotating joints was described in [Fisher, 1986] and [Fisher, 1989b]. To extend the work described here to recognise deformed objects with rotating parts only minor changes would be required. If rotation was specified to only occur between parts, the part selection and verification processes would remain unchanged since they are coordinate frame independent. Superpart selection could still be made by the verified existence of any parts, but the determination of the superpart parameters would be need to be modified. If three or more constituent parts had been identified, then with careful arrangement of the joints in the model the position and deformation parameters of a superpart could be determined independently of its parts. Resolving the stretch of a part relative to the stretch of its superpart would be simple once the rotation between the two had been determined by the verification process. Where only one or two rotating parts of a superpart had been identified then it might not be possible to exactly determine its position or deformation. The best that could be achieved would be a range of possible values that could be reduced as further correspondences were found during verification.

9.4.2 Limitations and Extensions in Model Selection

During model part selection, it is assumed that each of the three object surfaces in a combination belong to the same object part. However, this is not usually the case and a large amount of effort is wasted identifying those combinations which contain surfaces from different object parts. The inefficiencies caused by spurious data has been identified in other work [Grimson, 1989] [Grimson, 1990a] and the conclusion made that some method to identify the data subsets belonging to each object would be a great aid to efficiency. While such methods exist [Lowe, 1987], [Ballard, 1981] and [Jacobs, 1988] they proved unable to guarantee that all the elements of a data subset would belong to the same object without an excessive number of misinterpretations being generated [Grimson & Huttenlocher, 1988]. In the work described here, generating combinations of object surfaces from subsets of surfaces thought to belong to the same object, would greatly reduce the number of combinations generated and the number of interpretation trees required to interpret them. The incorrect inclusion in the subset of surfaces from different object parts should not present a problem provided there were three or more correct surfaces able to form a combination with a valid interpretation.

9.4.3 Limitations and Further Work in Model Verification

Probably the main limitation with this work concerns the identification of object features in the image during object verification. If a feature is represented in the model, in this case by a surface, then, unless it falls outside the image or faces away from the viewer, the model verification process expects it to be visible. Unfortunately this is not always the case and an object may not be visible in the image for other reasons:

1. Self–occlusion – An object surface is occluded, either partially or completely, by a surface or surfaces from part of the same object.

2. Occlusion – An object surface is occluded, either partially or completely, by a surface or surfaces from part of a different object.

3. Mis–segmentation – The original range data is segmented such that the surfaces representing an object have no correspondence to the surfaces represented in the model.

None of the problems described are unique to the problem of parameterised model matching but the changes in appearance exhibited by objects in a represented class exaggerate the problem.

If the orientation of an object in an image is known, the effects of self occlusion can be predicted by orientating the corresponding model and using a graphical technique such as "raycasting" [Foley & Van Dam, 1982] the points on its surfaces to see which are self–occluded. There is no reason why this should not be done here once the model had been deformed appropriately, but it would require the models to contain more exact surface descriptions, such as surface boundaries, than those currently used. The work in [Fisher, 1986] is a good example of such an approach.

Occlusion by a different object can be treated in a similar manner. As the position and orientation of each object is determined, then the deformed models can be transformed and a technique such as "z–buffering" [Foley & Van Dam, 1982] used to determine which surfaces would be visible to a viewer (in this case the camera). Once again a more exact surface description would be required. A similar approach was described in [Fisher, 1986].

The mis–segmentation of an object into surface patches bearing no resemblance to those represented in the model is probably the hardest problem of the three to overcome. Segmentation usually fails in one of two ways. Either an object surface is split into smaller patches, or two object surfaces are merged to form a larger surface. The problem can be reduced, as here, by modelling only those surfaces that are segmented with consistency. However, mis–segmentation will still occur. If sufficient features can be identified to select the correct model, then the surfaces in the model can be projected into the image and used to guide a "merge or split" procedure to produce surfaces matching those represented in the model. Once again a more exact representation of the model surfaces would be required than that used here.

All three problems would be overcome, or at least reduced, by the use of models with more accurate surface representations. Many techniques have been reviewed that can represent surfaces. Unfortunately, these are of limited use where the appearance of a surface can show such variation. A promising approach in character with the rest of this work might be the use of model surfaces that could dynamically bend and stretch to fit the object [Terzopoulos, 1983] [Terzopoulos *et al*, 1987b].

9.5 A Possible Parallel Implementation

The methods described in this thesis lend themselves to varying degrees to implementations on a parallel computer. The object description process uses purely local operations that could be simply and efficiently implemented on a four–way connected array processor such as the Distributed Array Processor (DAP) or a multiprocessor machine such as the Meiko transputer surface.

The generation and interpretation of combinations of object surfaces could also be achieved in parallel. Parallel implementations of interpretation tree searches [Flynn & Harris, 1985] have been achieved before on multiprocessor machines by assigning a different processor for each tree. Here, because an interpretation tree is generated for each combination, the number of trees is likely to be far greater. However, because the number of object surfaces in the image is known, the number of combinations that are to be generated and interpreted can be determined in advance and efficiently "farmed out" between the available processors of a multiprocessor machine.

Unfortunately such an implementation of the model selection process is not as efficient as it could be. The maximum depth of each interpretation tree is the same and since each combination is matched with the surfaces of all model parts, the maximum number of nodes in any tree is the same. Unfortunately the constraints used to identify incorrect matches can reduce the size of each interpretation tree to different degrees. This would mean some processors quickly rejecting all interpretations of each assigned combination while others might determine valid interpretations for all of the assigned combinations. The limited depth of each interpretation tree would mean that the effect of any pruning would be less than it would be on a traditional, larger tree but there is no simple solution to this problem. Careful sorting of the object surface combinations so that each processor gets an equal share of those separated by large distances (and less likely to have an interpretation) and those separated by more reasonable distances would help. Alternatively, some form of dynamic allocation as processors complete their assigned task, might prove more promising [Harris & Flynn, 1986].

The model parts selected are unlikely to be evenly distributed between the processors and for efficient use the selections made should be redistributed before model verification commences. While each model part can be verified in parallel, an efficient implementation of the process is difficult for the reasons given earlier. The problem is exaggerated because the depth of each search tree will vary with the number of surfaces in the model. This can be reduced to a degree by loading each processor with model parts containing roughly the same total number of model surfaces but will not overcome the problem completely. Again some form of dynamic task allocation might be promising.

Superpart selection and verification could be achieved in parallel in much the same manner as described previously.

Appendix A

List of Images

The range images depicted in this appendix are represented in one of two ways. Most are rendered using a false colour scheme. In these, those points that are closest to the viewer (camera) are represented in white. Those points further away are rendered in red through green to blue with dark blue representing those points furthest from the viewer. The black regions represent those areas in the image for which no range data is available. The alternative depiction is a cosine shaded representation using a grey scale representation. For these images the boundaries of the segmented surfaces are superimposed.

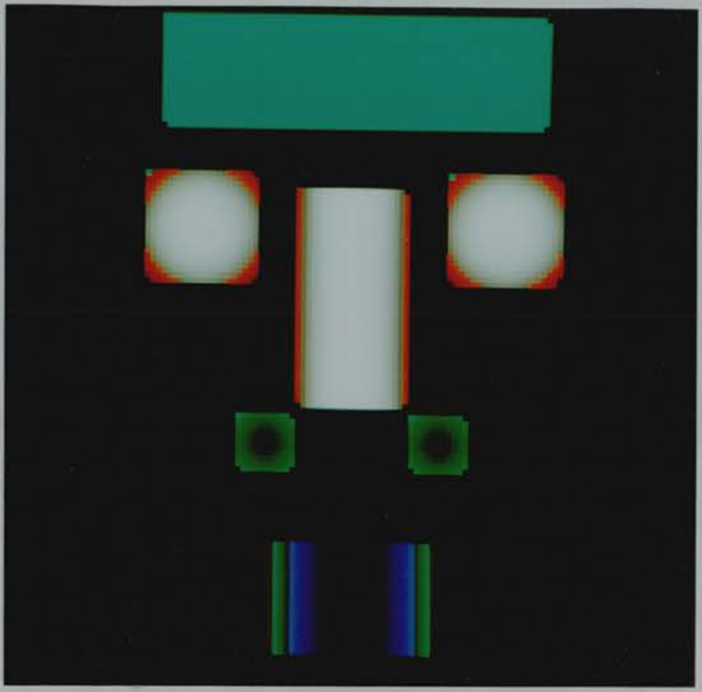


Figure A–1: The Basic Synthetic Face Object

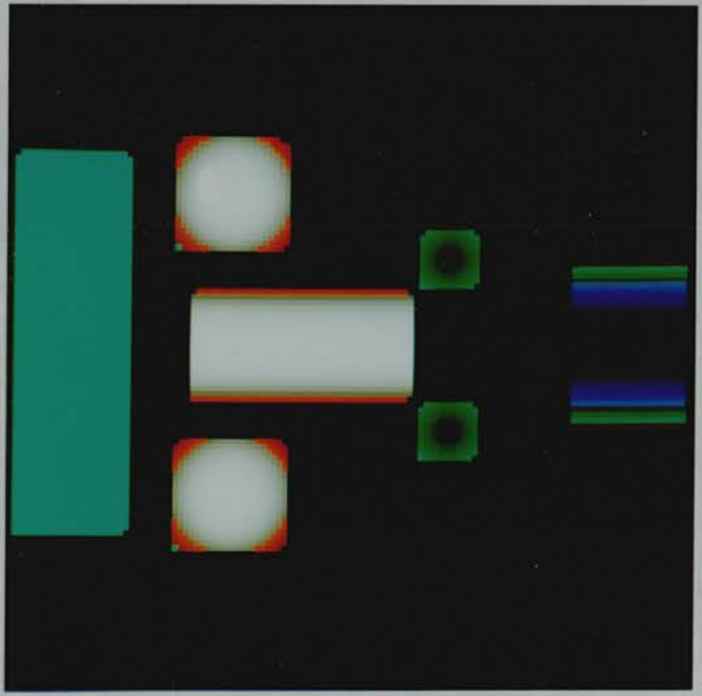


Figure A–2: The Rotated Synthetic Face Object

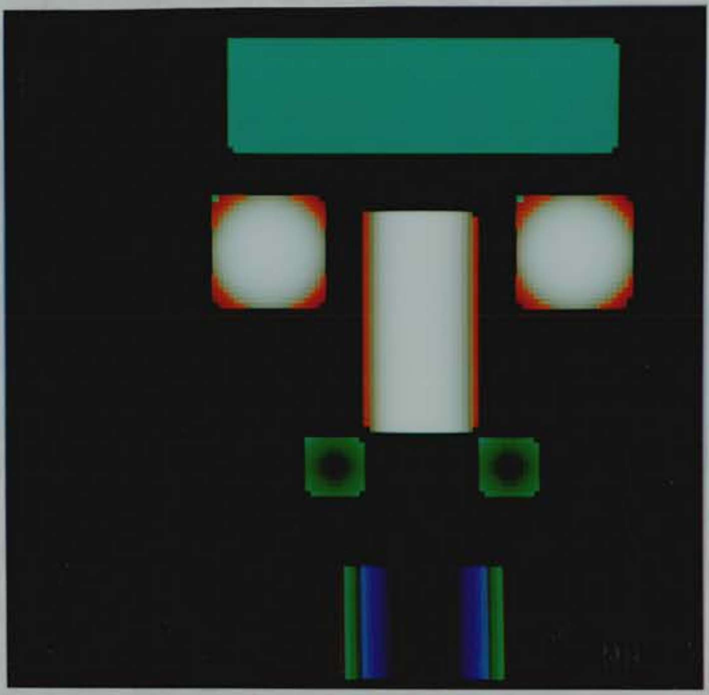


Figure A–3: The Translated Synthetic Face Object

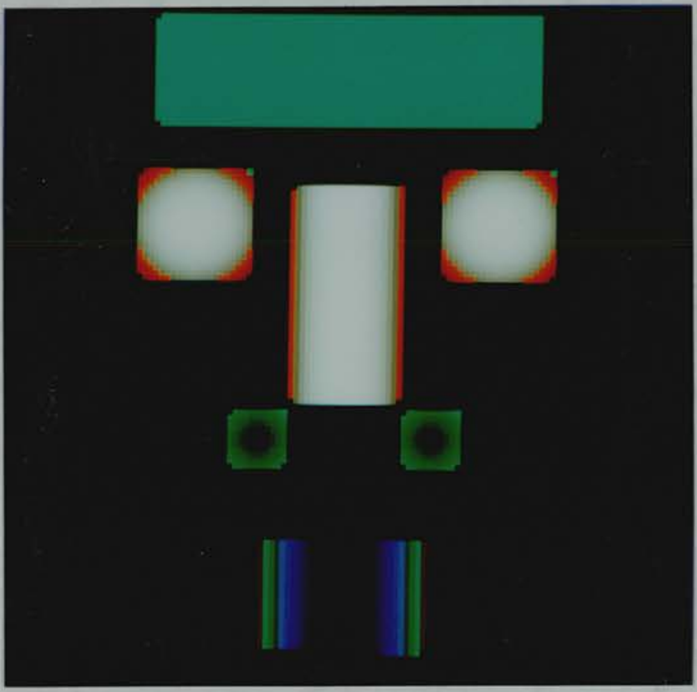


Figure A–4: The Scaled Synthetic Face Object

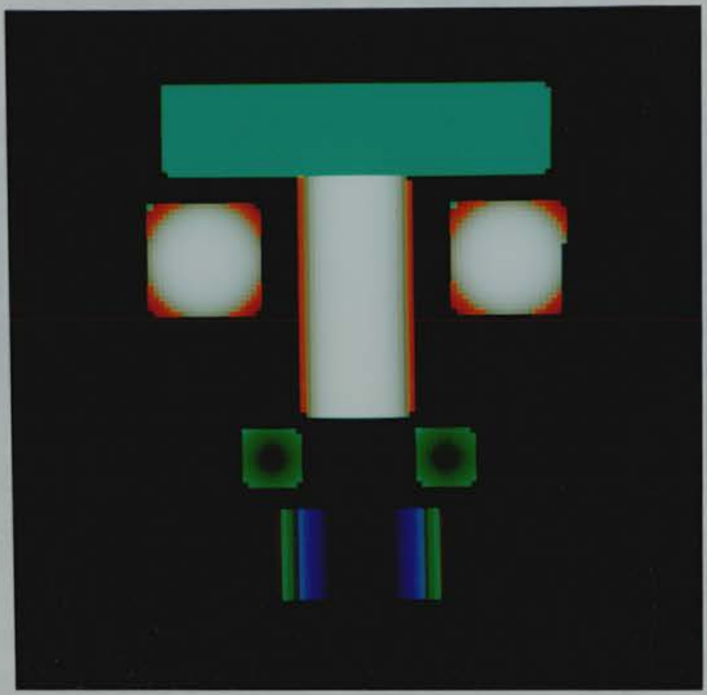


Figure A–5: The Synthetic Face Object Showing Stretching

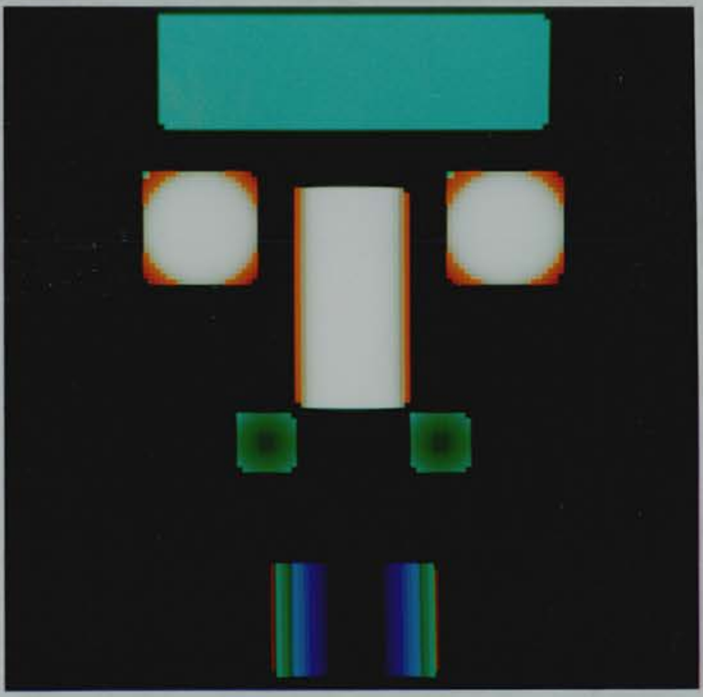


Figure A–6: The Synthetic Face Object Showing Variation

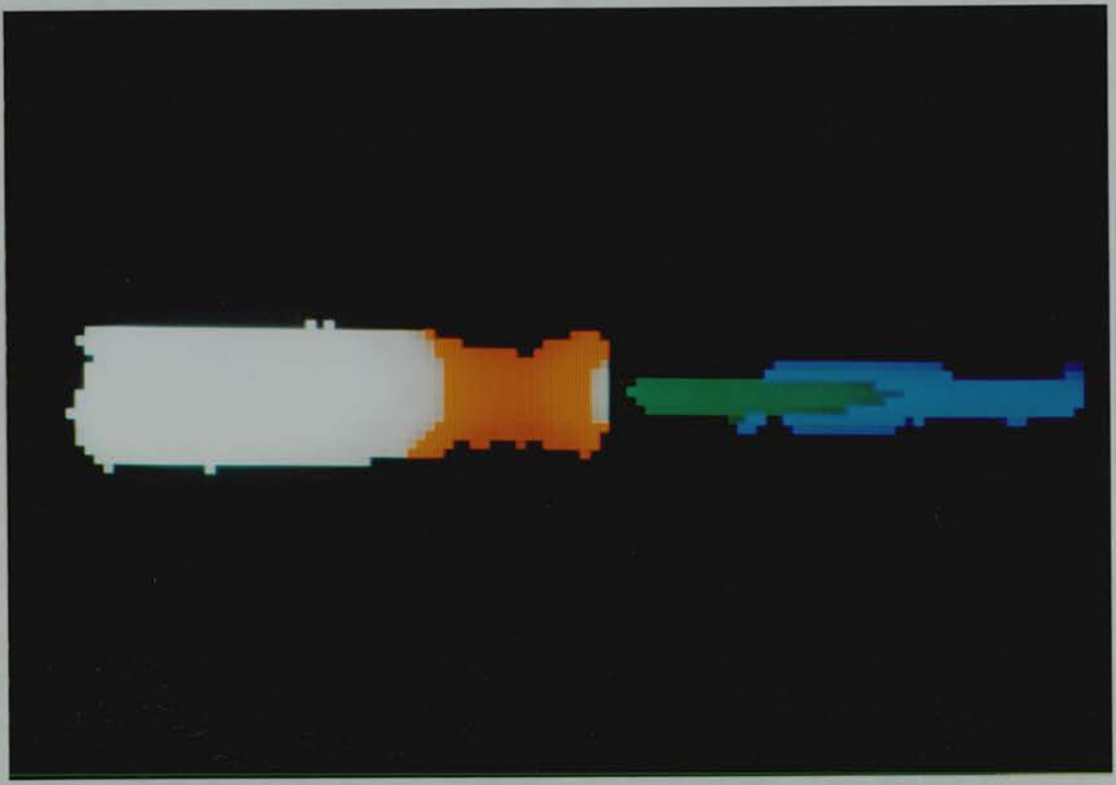


Figure A–7: The "Standard" Screwdriver

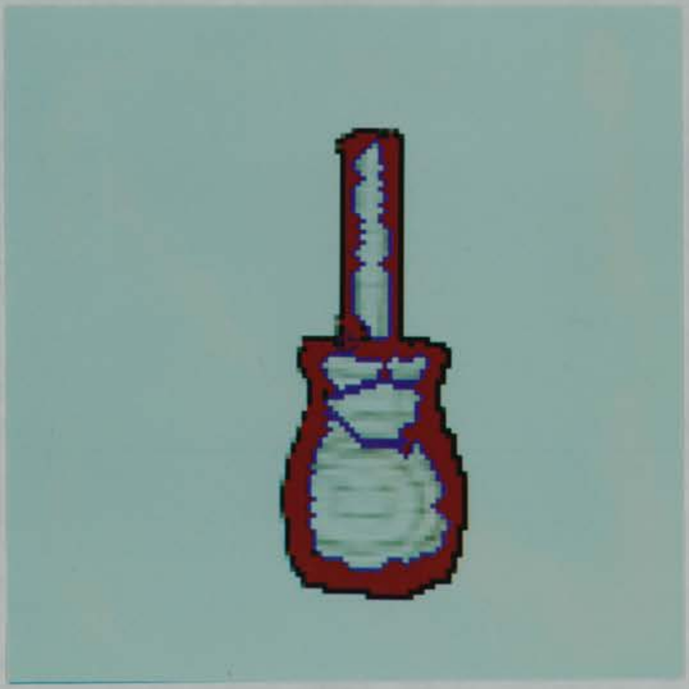


Figure A–8: The Small Screwdriver

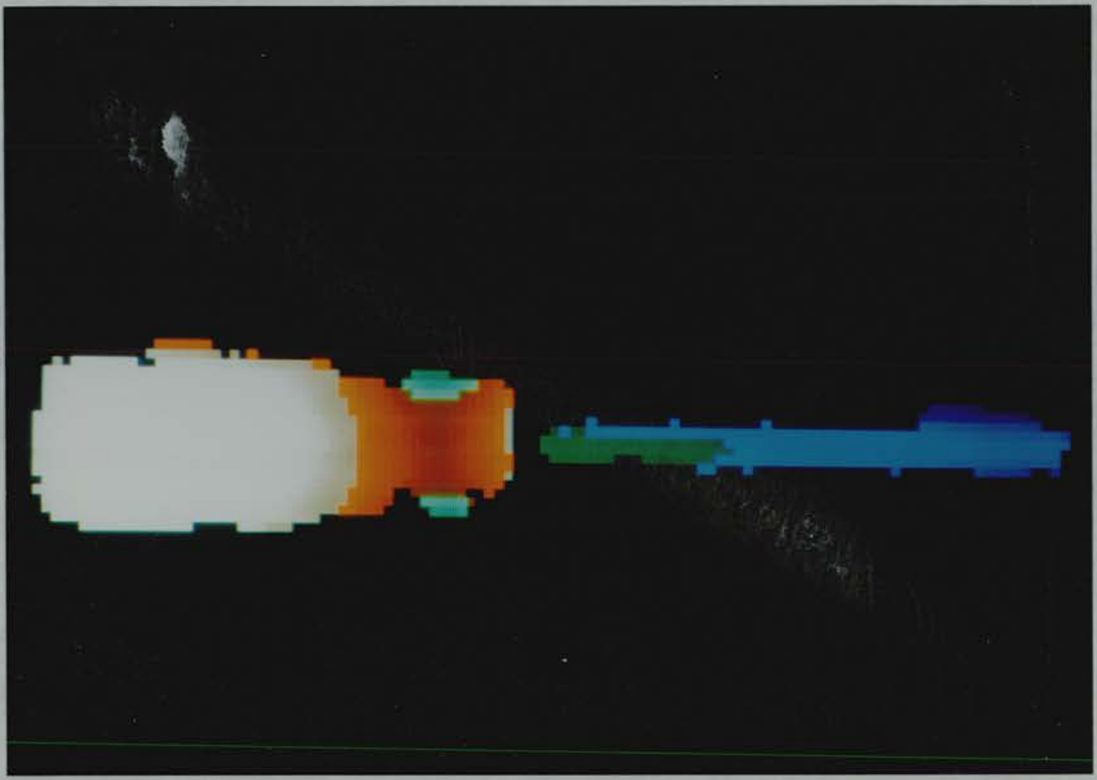


Figure A–9: The Large Screwdriver

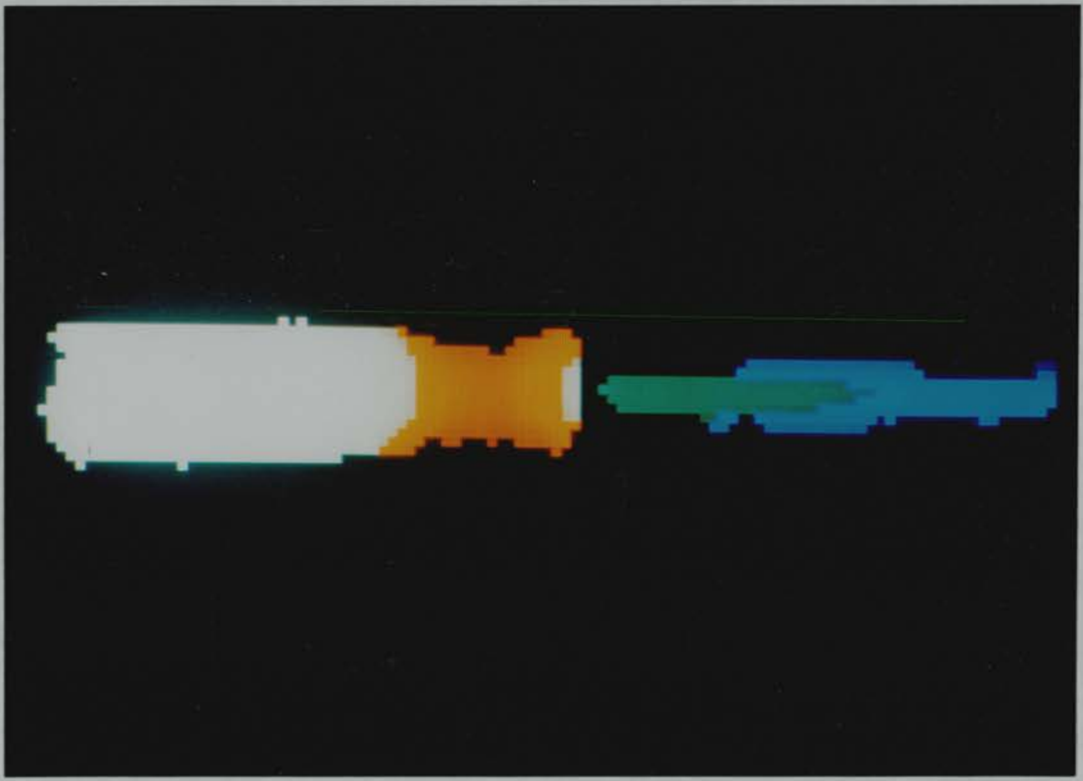


Figure A–10: The Scaled Up Standard Screwdriver

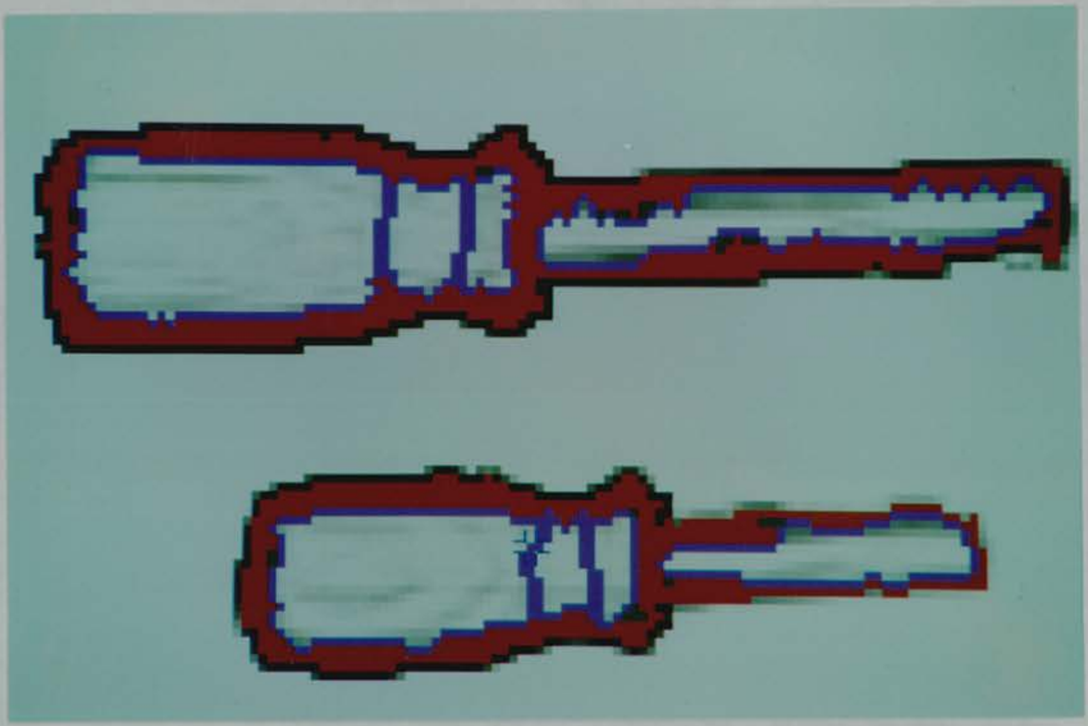


Figure A–11: Two Screwdrivers Exhibiting Different Scale and Stretch

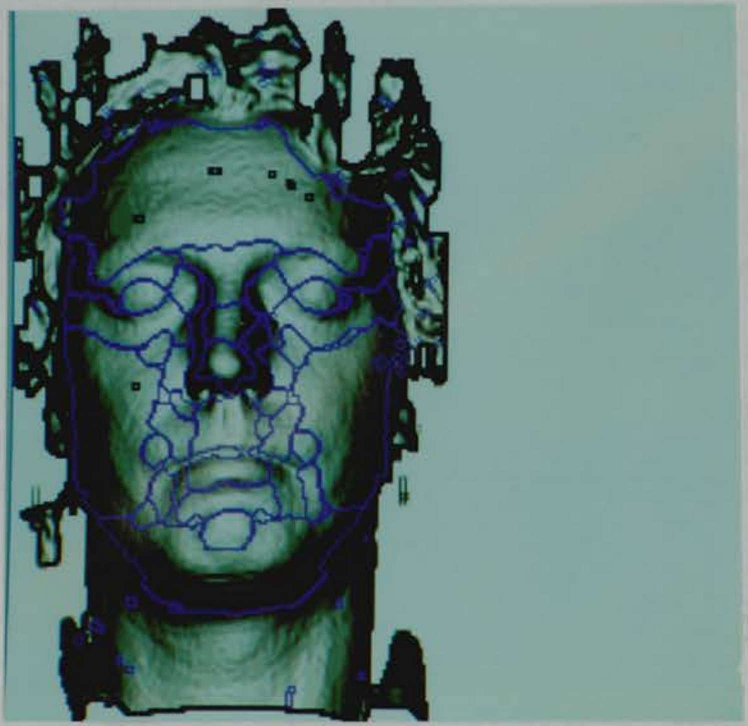


Figure A–12: An Adult Human Face

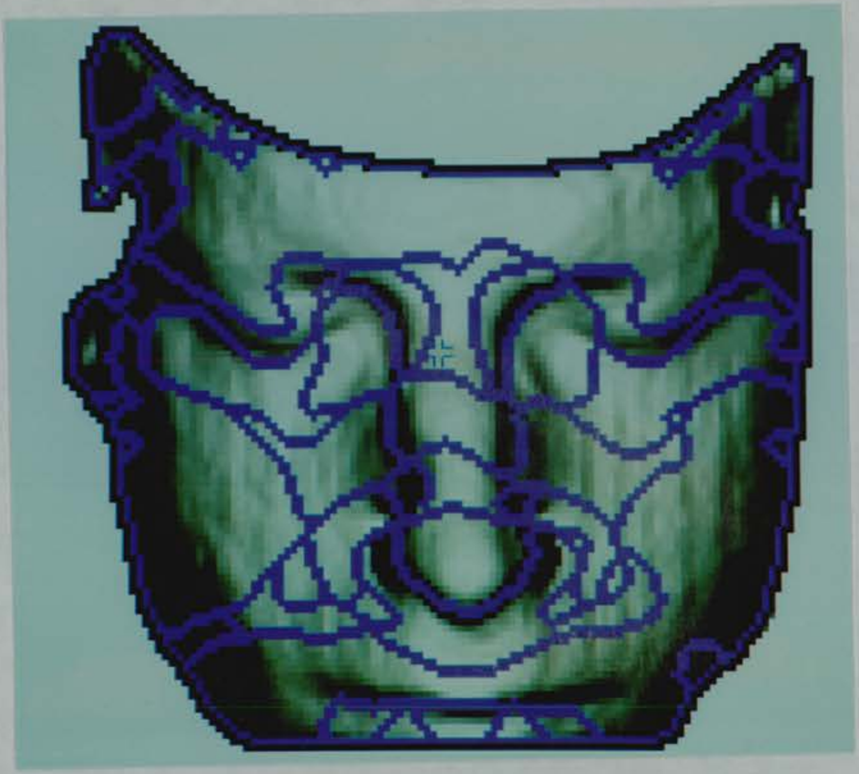


Figure A–13: A Second Adult Human Face

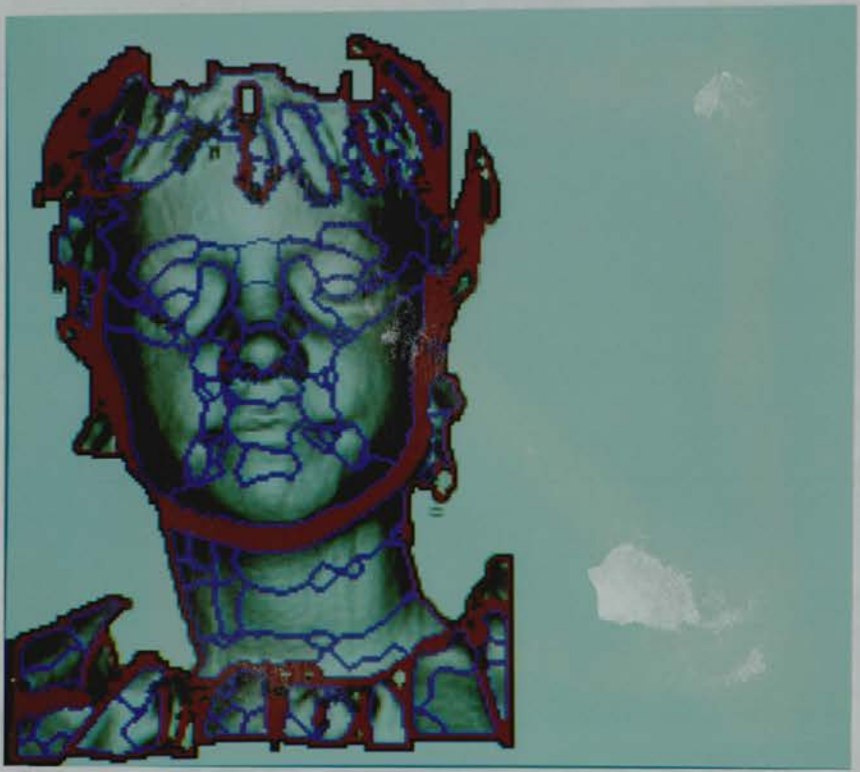


Figure A–14: A Child's Face

Appendix B

An Example Model (The Screwdriver)

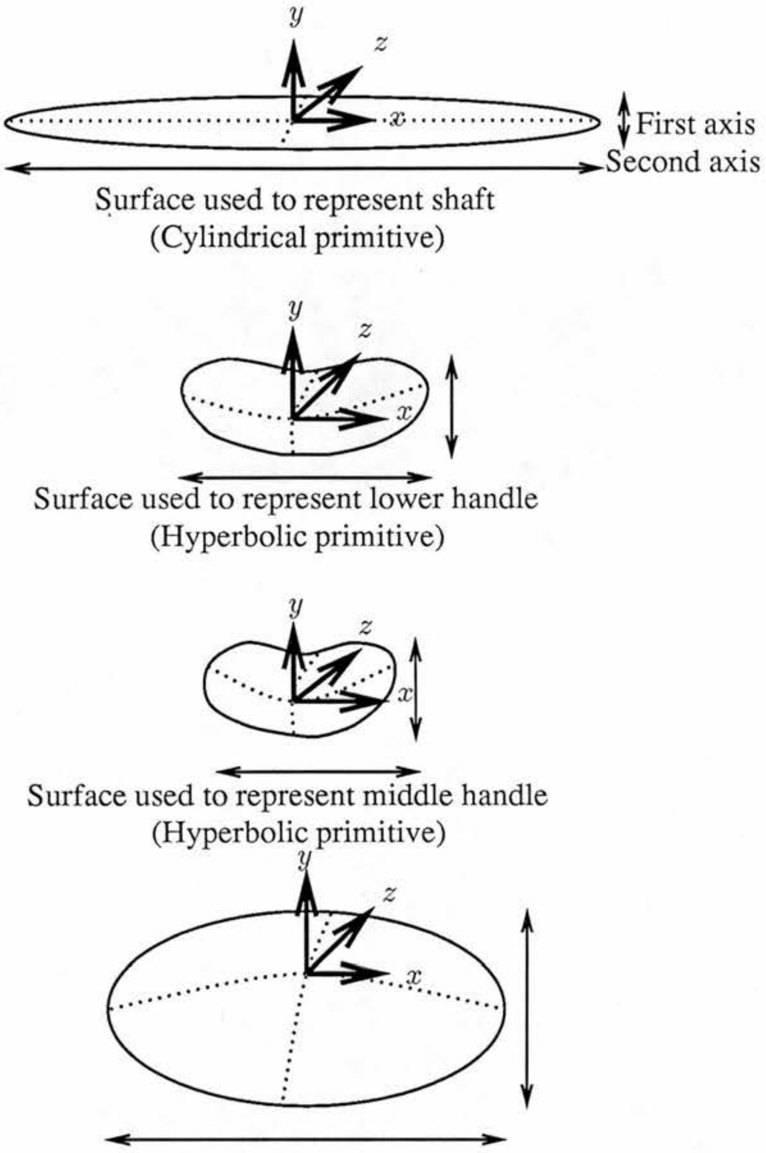
....

Descriptions of surface primitives 1–22 (all those *not* associated with screwdriver model) have been omitted for clarity.

....

```
% Surface primitive 23
curv_1 :  0.1587          % First principal curvature of
                          % surface ($mm^{-1}$)
curv_2 :  0.000           % Second principal curvature of
                          % surface ($mm^{-1}$)
curv_coeff :  1.0         % Weighting coefficient for any
                          % variation in curvature
diameter_1 :  1.0         % First axis of ellipsoid
                          % surface ($mms$) (unused here)
diameter_2 :  1.0         % Second axis of ellipsoid
                          % surface ($mms$) (unused here)
area :  98.0              % Area of surface ($mms^2$)
```

```
% Surface primitive 24
curv_1 :  0.0522            % First principal curvature of
                            % surface ($mm^{-1}$)
curv_2 :  -0.0019           % Second principal curvature of
                            % surface ($mm^{-1}$)
curv_coeff :  1.0           % Weighting coefficient for any
                            % variation in curvature
diameter_1 :  1.0           % First axis of ellipsoid
                            % surface ($mms$) (unused here)
diameter_2 :  1.0           % Second axis of ellipsoid
                            % surface ($mms$) (unused here)
area :  50.0                % Area of surface ($mms^2$)
% Surface primitive 25
curv_1 :  0.0512            % First principal curvature of
                            % surface ($mm^{-1}$)
curv_2 :  -0.0147           % Second principal curvature of
                            % surface ($mm^{-1}$)
curv_coeff :  1.0           % Weighting coefficient for any
                            % variation in curvature
diameter_1 :  1.0           % First axis of ellipsoid
                            % surface ($mms$) (unused here)
diameter_2 :  1.0           % Second axis of ellipsoid
                            % surface ($mms$) (unused here)
area :  25.0                % Area of surface ($mms^2$)
% Surface primitive 26
curv_1 :  0.0367            % First principal curvature of
                            % surface ($mm^{-1}$)
curv_2 :  0.0019            % Second principal curvature of
                            % surface ($mm^{-1}$)
curv_coeff :  1.0           % Weighting coefficient for any
                            % variation in curvature
diameter_1 :  1.0           % First axis of ellipsoid
                            % surface ($mms$) (unused here)
diameter_2 :  1.0           % Second axis of ellipsoid
                            % surface ($mms$) (unused here)
area :  540.0               % Area of surface ($mms^2$)
```



Surface used to represent shaft
(Cylindrical primitive)

Surface used to represent lower handle
(Hyperbolic primitive)

Surface used to represent middle handle
(Hyperbolic primitive)

Surface used to represent upper handle
(Positive elliptical primitive)

Although there exists a facility to specify the dimensions of each surface primitive it is currently unused. However, the areas of the surfaces *are* specified and the dimensions of the surfaces shown here are intended to reflect this.

Although the coordinate frame of each surface primitive is represented, the position and orientation of each primitive is *not* specified at this stage.

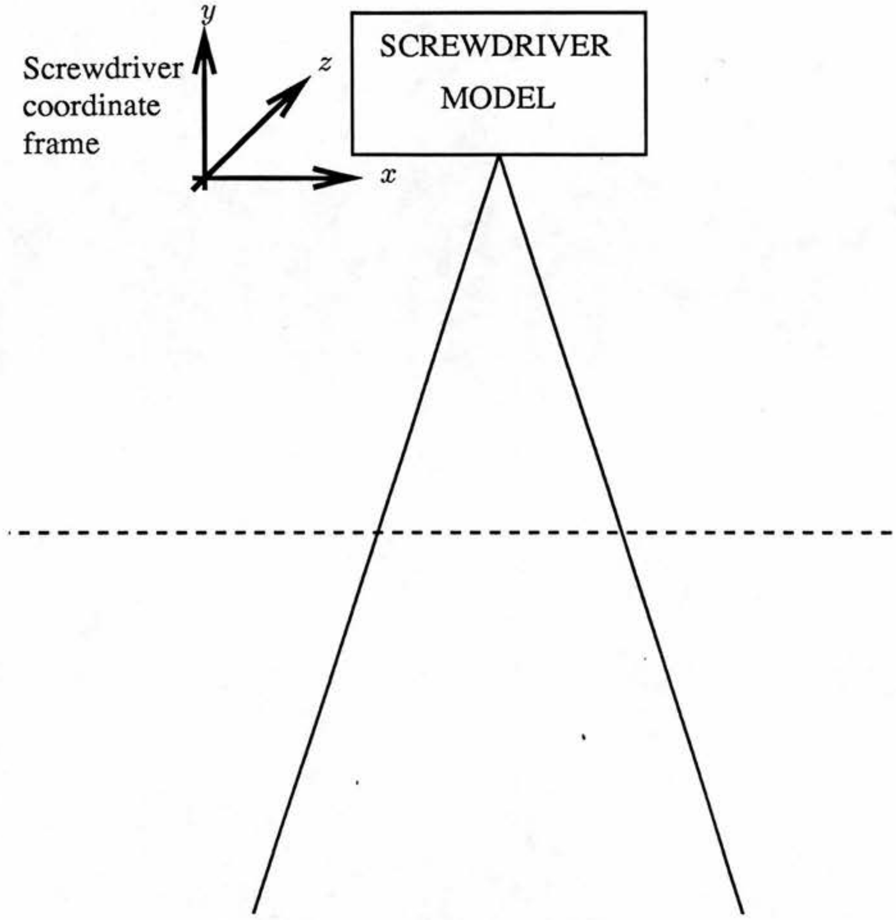
Figure B–1: The Surface Primitives Used by the Screwdriver Model

.....

Full descriptions of all the models start here but here all but the screwdriver model are omitted for clarity.

.....

```
% Screwdriver model
Name :  SCREWDRIVER                                 % Name of model
Model type :  Model                                 % Denotes top level of
                                                    % model description
Scale lower limit :  0.3                            % Lower limit of
                                                    % global scale $S_g$
Scale upper limit :  3.0                            % Upper limit of
                                                    % global scale $S_g$
Stretch lower limits :  0.75 0.75 0.75              % Lower limits of
                                                    % stretch $S_x$, $S_y$ and $S_z$
Stretch upper limits :  1.50 1.50 1.50              % Upper limits of
                                                    % stretch $S_x$, $S_y$ and $S_z$
Posn coeffs :  1.0 1.0 1.0                          % Weighting coefficient for
                                                    % any variation in position
Model position :  0.0 0.0 0.0                       % Position of model
                                                    % $(x, y, z)$ $(mms)$ (not used)
Orient coeffs :  0.8 0.8 0.8                        % Weighting coeff.  for any
                                                    % variation in orientation
Model orientation :  0.0 0.0 0.0                    % Orientation of model
                                                    % $(r, s, t)$ $(radians)$ (not used)
Number of parts :  2                                % Number of parts (in this
                                                    % case parts) in this model
```

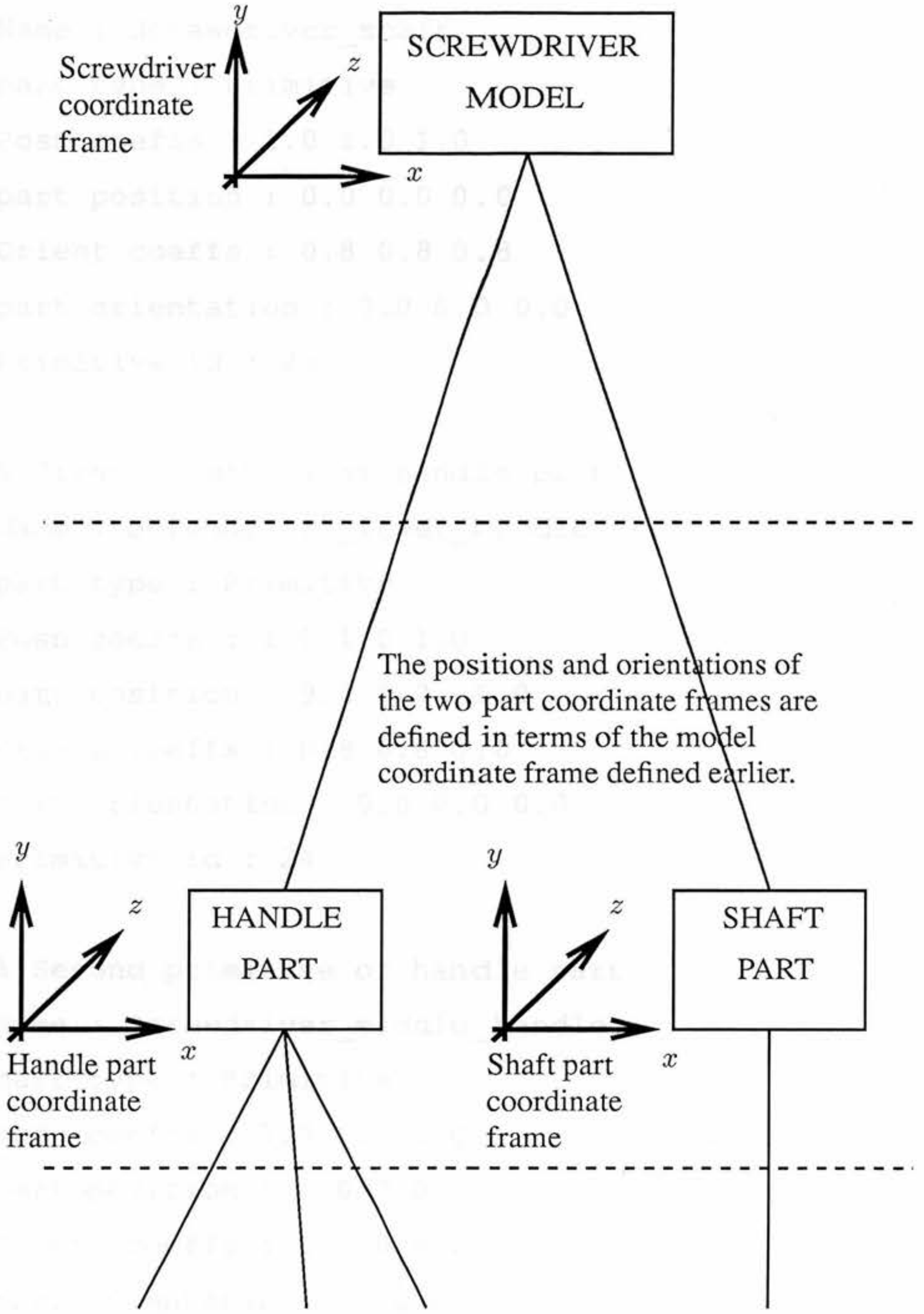


The screwdriver model is defined as
having two parts which may be superparts,
parts or primitives but at this level
the type is not yet defined.

The model coordinate frame performs no
function at this stage.

Figure B–2: The Screwdriver Model Description

```
% First part of model
Name :  SCREWDRIVER_SHAFT                    % Name of part
part type :  Part                            % Denotes description
                                             % of model part
Stretch lower limits :  0.75 0.75 0.75       % Lower limits of
                                             % stretch $S_x$, $S_y$ and $S_z$
Stretch upper limits :  1.50 1.50 1.50       % Upper limits of
                                             % stretch $S_x$, $S_y$ and $S_z$
Posn coeffs :  1.0 1.0 1.0                   % Weighting coefficient for
                                             % any variation in position
part position :  53.3 0.0 5.1                % Position of part coordinate
                                             % frame $x$, $y$ and $z$ $(mms)$
                                             % in model coordinate frame
Orient coeffs :  0.8 0.8 0.8                 % Weighting coefficient for
                                             % any variation in orientatio
part orientation :  0.0 0.0 0.0              % Orientation of part
                                             % $(r, s, t)$ $(radians)$
                                             % in model coordinate frame
Number of parts :  1                         % Number of parts
                                             % (surface primitives here)
                                             % within this part.
% Second part of model
Name :  SCREWDRIVER_HANDLE              % Name of part
part type :  Part                       % Denotes description
                                        % of model part
Stretch lower limits :  0.75 0.75 0.75  % Lower limits of
                                        % stretch $S_x$, $S_y$ and $S_z$
Stretch upper limits :  1.50 1.50 1.50  % Upper limits of
                                        % stretch $S_x$, $S_y$ and $S_z$
Posn coeffs :  1.0 1.0 1.0              % Weighting coefficient for
                                        % any variation in position
part position :  0.0 0.0 0.0            % Position of part coordinate
                                        % frame $x$, $y$ and $z$ ($mms$)
                                        % in model coordinate frame
Orient coeffs :  0.8 0.8 0.8            % Weighting coefficient for
                                        % any variation in orientation
part orientation :  0.0 0.0 0.0         % Orientation of part
                                        % $(r, s, t)$ ($radians$)
                                        % in model coordinate frame
Number of parts :  3                    % Number of parts
                                        % (surface primitives here)
                                        % within this part.
```


y
Screwdriver coordinate frame
z
x
SCREWDRIVER MODEL
y
z
HANDLE PART
x
Handle part coordinate frame
y
z
SHAFT PART
x
Shaft part coordinate frame

The positions and orientations of the two part coordinate frames are defined in terms of the model coordinate frame defined earlier.

The handle part consists of three parts

The shaft part consists of one part

Figure B–3: The Screwdriver Parts Descriptions

```
% Only primitive of shaft part
Name :  Screwdriver_shaft         % Name of only surface primitive
                                  % in shaft part
part type :  Primitive            % Denotes description of
                                  % model surface primitive
Posn coeffs :  1.0 1.0 1.0        % Weighting coefficient for
                                  % any variation in position
part position :  0.0 0.0 0.0      % Position of part coordinate
                                  % frame x, y and z (mms)
                                  % within model coordinate frame
Orient coeffs :  0.8 0.8 0.8      % Weighting coefficient for
                                  % any variation in orientation
part orientation :  0.0 0.0 0.0   % Orientation of part
                                  % (r,s,t) (radians)
                                  % within model coordinate frame
Primitive id :  23                % Number of primitive
                                  % representing this surface
% First primitive of handle part
Name :  Screwdriver_lower_handle  % Name of first surface primitive
                                  % in handle part
part type :  Primitive            % Denotes description of
                                  % model surface primitive
Posn coeffs :  1.0 1.0 1.0        % Weighting coefficient for
                                  % any variation in position
part position :  9.0 0.0 -1.0     % Position of part coordinate
                                  % frame x, y and z (mms)
                                  % within part coordinate frame
Orient coeffs :  0.8 0.8 0.8      % Weighting coefficient for
                                  % any variation in orientation
part orientation :  0.0 0.0 0.0   % Orientation of part
                                  % (r,s,t) (radians)
                                  % within model coordinate frame
Primitive id :  24                % Number of primitive
                                  % representing this surface
% Second primitive of handle part
Name :  Screwdriver_middle_handle  % Name of second surface primitive
                                   % in handle part
part type :  Primitive             % Denotes description of
                                   % model surface primitive
Posn coeffs :  1.0 1.0 1.0         % Weighting coefficient for
                                   % any variation in position
part position :  0.0 0.0 0.0       % Position of primitive coordinate
                                   % frame x, y and z (mms)
                                   % within part coordinate frame
Orient coeffs :  0.8 0.8 0.8       % Weighting coefficient for
                                   % any variation in orientation
part orientation :  0.0 0.0 0.0    % Orientation of primitive
                                   % (r,s,t) (radians)
                                   % within part coordinate frame
Primitive id :  25                 % Number of primitive
                                   % representing this surface
% Third primitive of handle part
Name :  Screwdriver_upper_handle   % Name of third surface primitive
                                   % in handle part
part type :  Primitive             % Denotes description of
                                   % model surface primitive
Posn coeffs :  1.0 1.0 1.0         % Weighting coefficient for
                                   % any variation in position
part position :  -32.0 0.0 -4.5    % Position of primitive coordinate
                                   % frame x, y and z (mms)
                                   % within part coordinate frame
Orient coeffs :  0.8 0.8 0.8       % Weighting coefficient for
                                   % any variation in orientation
part orientation :  0.0 0.0 0.0    % Orientation of primitive
                                   % (r,s,t) (radians)
                                   % within part coordinate frame
Primitive id :  26                 % Number of primitive
                                   % representing this surface
```

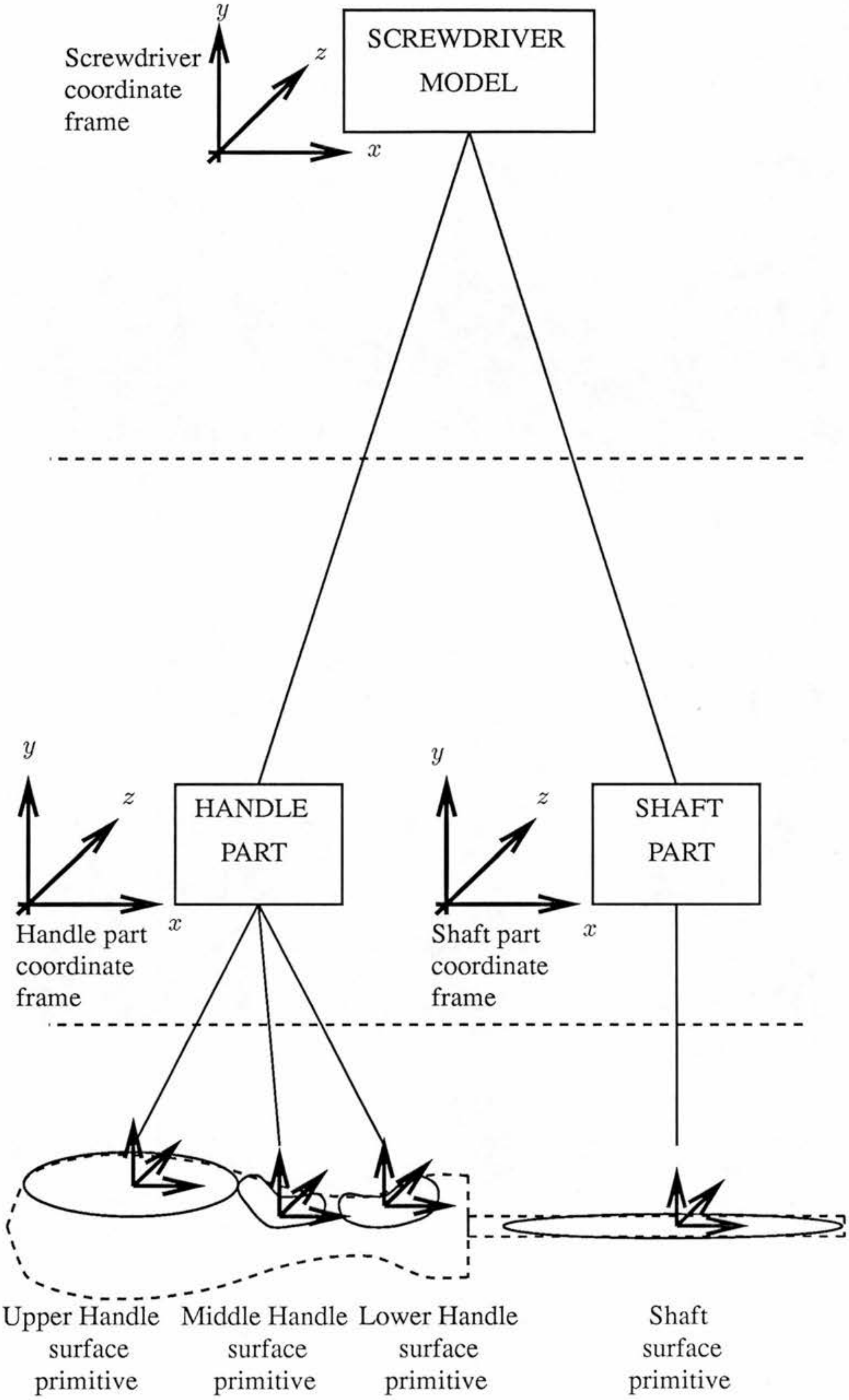

Screwdriver coordinate frame
y
z
x
SCREWDRIVER MODEL
y
z
HANDLE PART
x
Handle part coordinate frame
y
z
SHAFT PART
x
Shaft part coordinate frame
Upper Handle surface primitive
Middle Handle surface primitive
Lower Handle surface primitive
Shaft surface primitive

Figure B–4: The Full Screwdriver Model Descriptions

Bibliography

[Ayache & Faugeras, 1986] Ayache, N.J. and Faugeras, O.D. (1986). HYPER: A new approach for the recognition and positioning of two dimensional objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):44–54.

[Bajcsy & Solina, 1987] Bajcsy, R. and Solina, F. (1987). Three dimensional object representation revisited. In *First International Conference on Computer Vision*, pages 231–240.

[Ballard, 1981] Ballard, D.H. (1981). Generalising the Hough transform to detect arbitrary patterns. *Pattern Recognition*, 13:111–122.

[Barr, 1981] Barr, A.H. (1981). Superquadrics and angle–preserving transformations. *IEEE Computer Graphics and Application*, 1:1–20.

[Bolles & Cain, 1982] Bolles, R.C. and Cain, R.A. (1982). Recognizing and locating partially visible objects, the local feature focus method. *The International Journal of Robotics Research*, 1(3):57–82.

[Bolles & Horaud, 1986] Bolles, R.C. and Horaud, P. (1986). 3DPO: A three–dimensional part orientation system. *The In-*

ternational Journal of Robotics Research, 5(3):3–26.

[Brooks, 1981] Brooks, R. (1981). *Model-Based Computer Vision*. UMI.

[Brooks, 1983] Brooks, R.A. (1983). Model based three dimensional interpretations of two dimensional images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5:140–149.

[Cai, 1990] Cai, L.D. (1990). *Scale–Based Surface Understanding Using Diffusion Smoothing*. Unpublished Ph.D. thesis, Department of Artificial Intelligence, University of Edinburgh.

[Coons, 1974] Coons, S.A. (1974). Surface patches and b–spline curves. In Barnhill, R.E. and Riesenfeld, R.F., (eds.), *Computer Aided Geometric Design*. Academic Press New York.

[Drumheller, 1987] Drumheller, M. (1987). Mobile robot localization using sonar. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(2):325–332.

[Faugeras & Hebert, 1983] Faugeras, O.D. and Hebert, M. (1983). A 3–D recognition and positioning algorithm using geometrical matching between primitive surfaces. In *Eighth International Joint Conference on Artificial Intelligence*, pages 996–1002.

[Fischler & Elschlager, 1973] Fischler, M.A. and Elschlager, R.A. (1973). The representation and matching of pictorial structures. *IEEE Transactions on Computing*, 22:67–92.

[Fisher & Orr, 1987] Fisher, R.B. and Orr, M.J.L. (1987). Solving geometric constraints in a parallel network. In *Proceedings of the Third Alvey Vision Conference*, pages 87–95.

[Fisher, 1985] Fisher, R.B. (1985). SMS:a suggestive modelling system for object recognition. Working Paper 185, Department of Artificial Intelligence, University of Edinburgh.

[Fisher, 1986] Fisher, R.B. (1986). *From Surfaces to Objects: Recognising Objects Using Surface Information and Object Models*. Unpublished Ph.D. thesis, Department of Artificial Intelligence, University of Edinburgh.

[Fisher, 1989a] Fisher, R.B. (1989a). *From Surfaces to Objects: Computer Vision and Three Dimensional Scene Analysis.*, pages 128–133. John Wiley and Sons.

[Fisher, 1989b] Fisher, R.B. (1989b). *From Surfaces to Objects: Computer Vision and Three Dimensional Scene Analysis*. John Wiley and Sons.

[Flynn & Harris, 1985] Flynn, A.M. and Harris, J.G. (1985). Recognition algorithms for the connection machine. In *International Joint Conference on Artificial Intelligence*, pages 57–60.

[Foley & Van Dam, 1982] Foley, J.D. and Van Dam, A. (1982). *Fundamentals of Interactive Computer Graphics*. Addison–Wesley.

[Gaston & Lozano-Pérez, 1984] Gaston, P.C. and Lozano-Pérez, T. (1984). Tactile recognition and localization using object models:the case of polyhedra on a plane. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:257–266.

[Goad, 1985] Goad, C. (1985). Fast 3–d model based vision. In Pentland, A.P., (ed.), *From Pixels to Predicates*, pages 371–391. Norwood, N.J. Ablex Pub. Corp.

[Gordon & Riesenfeld, 1974] Gordon, W.J. and Riesenfeld, R.F. (1974). B–spline curves and surfaces. In Barnhill, R.E. and Riesenfeld, R.F., (eds.), *Computer Aided Geometric Design*. Academic Press New York.

[Grimson & Huttenlocher, 1988] Grimson, W.E.L. and Huttenlocher, D.P. (1988). On the sensitivity of the Hough transform for object recognition. In *Second International Conference on Computer Vision*, pages 700–706.

[Grimson & Lozano-Pérez, 1984] Grimson, W.E.L. and Lozano-Pérez, T. (1984). Model-based recognition and localisation from sparse range or tactile data. *The International Journal of Robotics Research*, 3(3):3–35.

[Grimson & Lozano-Pérez, 1987] Grimson, W.E.L. and Lozano-Pérez, T. (1987). Localizing overlapping parts by searching the interpretation tree. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9:469–482.

[Grimson, 1987] Grimson, W.E.L. (1987). Recognition of object families using parameterised models. In *First International Conference on Computer Vision*, pages 93–101.

[Grimson, 1988] Grimson, W.E.L. (1988). The combinatorics of object recognition in cluttered environments using constrained search. In *Second International Conference on Computer Vision*, pages 218–227.

[Grimson, 1989] Grimson, W.E.L. (1989). The combinatorics of heuristic search termination for object recognition in cluttered environments. AI Memo 1111, MIT.

[Grimson, 1990a] Grimson, W.E.L. (1990a). The combinatorics of heuristic search termination for object recognition in cluttered environments. In *Proceedings of European Conference on Computer Vision*, pages 552–556.

[Grimson, 1990b] Grimson, W.E.L. (1990b). *Object Recognition by Computer*. The MIT Press.

[Hanson & Riseman, 1978] Hanson, A.R. and Riseman, E.M. (1978). Visions:a computer system for interpreting scenes. In Hanson, A.R. and Riseman, E.M., (eds.), *Computer Vision Systems*, pages 303–333. Academic Press.

[Harris & Flynn, 1986] Harris, J.G. and Flynn, A.M. (1986). Object recognition using the connection machines router. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*

[Huttenlocher & Ullman, 1987] Huttenlocher, D.P. and Ullman, S. (1987). Object recognition using alignment. In *First International Conference on Computer Vision*, pages 102–111.

[Ikeuchi, 1987] Ikeuchi, K. (1987). Generating an interpretation tree from a CAD model for 3D–object recognition in bin–picking tasks. *International Journal of Computer Vision*, 1(2):145–165.

[Jacobs, 1988] Jacobs, D.W. (1988). The use of grouping in visual object recognition. AI Memo 1023, MIT.

[Knapman, 1987] Knapman, J. (1987). 3D model identification from stereo data. In *First International Conference on Computer Vision*, pages 547–551.

[Lowe, 1987] Lowe, D.G. (1987). Three–dimensional object recognition from single two–dimensional images. *Artificial Intelligence*, 31:355–395.

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

[Lozano-Pérez *et al*, 1987] Lozano-Pérez, T., Jones, J.L., Mazer, E., O'Donnell, P.A. and Grimson, W.E.L. (1987). Handey: A robot system that recognises, plans and manipulates. In *IEEE International Conference on Robotics and Automation*, pages 843–849.

[Marr, 1982] Marr, D. (1982). *Vision*. W.H.Freeman and Co.

[Murray & Cook, 1988] Murray, D.W. and Cook, D.B. (1988). Using the orientation of fragmentary 3D edge segments for polyhedral object recognition. *International Journal of Computer Vision*, 2:153–169.

[Murray, 1987a] Murray, D.W. (1987a). Model–based recognition using 3–D shape alone. *Computer Vision, Graphics and Image Processing*, 40:250–266.

[Murray, 1987b] Murray, D.W. (1987b). Model–based recognition using 3–D structure from motion. *Image and Vision Computing*, 5:85–90.

[Naidu & Fisher, 1990] Naidu, D.K. and Fisher, R.B. (1990). Range sensor hardware description. Imagine Project 23, Edinburgh.

[Naidu *et al*, 1990] Naidu, D.K., Fisher, R.B. and Cameron, G. (1990). User guide for the laser image acquisition package. Imagine Project 18, Edinburgh.

[Pentland, 1985] Pentland, A.P. (1985). The parts of perception. In Pentland, A.P., (ed.), *From Pixels to Predicates*, pages 121–176. Norwood, N.J. Ablex Pub. Corp.

[Pentland, 1986] Pentland, A.P. (1986). Parts:structured description of shape. In *Proceedings of American Association of Artificial Intelligence*, pages 695–701.

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

[Press *et al*, 1988] Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T. (1988). *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press.

[Reid, 1990] Reid, I. (1990). Recognition of parameterised models from 3D data. In *IAPR Workshop on Machine Vision Applications*, pages 347–350.

[Rioux & Cournoyer, 1988] Rioux, M. and Cournoyer, L. (1988). *The NRCC Three–Dimensional Image Data Files*. National Research Council of Canada.

[Roberts, 1965] Roberts, L.G. (1965). Machine perception of three–dimensional solids. *Optical and Electro–Optical Information Processing*, 2:159–197.

[Simsarian *et al*, 1990] Simsarian, K.T., Nandhakumar, N. and Olson, T.J. (1990). Mobile robot self–localization from range data using view–invariant regions. In *Proceedings of the Fifth IEEE International Symposium on Intelligent Control*.

[Terzopoulos, 1983] Terzopoulos, D. (1983). Multilevel computational processes for visual surface reconstruction. *Computer Vision, Graphics and Image Processing*, 24:52–96.

[Terzopoulos *et al*, 1987a] Terzopoulos, D., Platt, J., Barr, A. and Fleischer, K. (1987a). Elastically deformable models. *Computer Graphics*, 21(4):205–214.

[Terzopoulos *et al*, 1987b] Terzopoulos, D., Witkin, A. and Kass, M. (1987b). Symmetry–seeking models for 3D object recognition. In *First International Conference on Computer Vision*, pages 269–276.

[Weatherburn, 1931] Weatherburn, C.E. (1931). *Differential Geometry ofThree Dimensions*. Cambridge University Press.