Representation, Feature Extraction and Geometric Constraints for Recognising 3D Objects from a Single Perspective View

by

Kok Cheong WONG

Submitted in

Partial Fulfillment of the

Doctor of Philosophy

of

University of Surrey

United Kingdom

November 1992
Dedicated to

Hock Mui Lee, my parents, brother

and

Dr. S. E. Williamson
Acknowledgements

I offer a special note of thanks to my supervisor, Professor Josef Kittler, for his encouragement and thoughtful comments. His serious dedication has been a true inspiration to me. I am extremely grateful to Josef for giving me an opportunity to develop my own ideas in almost all the work presented in this thesis. The author would like to express his thanks to the Overseas Research Scheme Award from United Kingdom and ESPRIT for their financial support.

It is a pleasure to record my particular thanks to Dr. S. E. Williamson. To Syd, I owe a special debt of gratitude, for his support and encouragement.

In the preparation of the Chapter 5 (Recognition of Polyhedral Objects Using Triangle-pair Features) of this thesis, I have benefited considerably from the valuable advice and criticism of Professor Cheng Yu.

Over the years, I have received useful comments and generous advice from many people in the University of Surrey. In particular, I thank Graeme Jones, John Illingworth, Geoff Nicholls, J. H. B. Deane, P. J. Aston, Hei-Fat Isaac Ng, A. Etemadi, J-P. Schmidt, George Matas, Dekun Yang, Maria Petrou, S Ragupathi, Paolo Remagnino, Edward Chilton and Paul Hoad. I deeply appreciate their cheerful friendship.

Finally, I would like to express my greatest love and gratitude to my wife, Hock Mui Lee, to my parents and brother who are making my life in general a more enjoyable and challenging experience. I sincerely thank them for giving me their support in all aspects, more than I often deserved.
Representation, Feature Extraction and Geometric Constraints for Recognising 3D Objects from a Single Perspective View

by

K. C. Wong

Submitted to the
Department of Electronic & Electrical Engineering,
University of Surrey, England
November 5, 1992

Abstract

This dissertation considers the problem of modelling, feature extraction and recognizing 3D objects from a single perspective view. A solid modelling system based on generalized cylinder is presented. A new algorithm is proposed for grouping 2D line segments into intermediate token features to be used as geometric cues for indexing into the model database and for generating hypotheses for polyhedral objects. A polyhedral object recognition system using a hypothesis and verification paradigm has been proposed and developed.

In the modelling system, generalized cylinders are used as geometric primitives for representing objects. The analysis of generalized cylinders is presented. A number of useful expressions and properties of the contour generators of straight homogeneous generalized cylinders are derived under perspective projection. Right and oblique straight homogeneous generalized cylinders with circular and arbitrary cross-section are discussed.

A novel algorithm for extracting geometric features such as triples of connected edges, triangle-pairings, image trihedral vertices and closed polygons is implemented. Both heuristic and physical rules are utilised to control the combinatorial explosion of the feature grouping process. Physical rules are used to reject closed polygons which are incompatible with a single planar surface hypothesis. Experiments are demonstrated on real data and many features which could reasonably be due to spatial physical properties of the objects are identified. Only a few spurious features are accidentally detected. These irrelevant features are then pruned away in the hypothesis generation and verification process modules of the proposed recognition system.

A polyhedral object recognition system based on a single perspective image is developed. A hypothesis and verification paradigm based on the use of local geometric features of objects is presented. In the framework, two high-level geometric primitives, namely triangle-pair and quadrilateral are employed as key features for model invocation and hypothesis generation. Two geometric constraints, namely distance and angle constraints are proposed and integrated into the recognition system. Many
model and scene correspondences are pruned away in the early stage of the matching process using the two geometric constraints. As a by-product of the hypothesis generation the relative pose of the 3D objects expressed in camera frame is recovered. A verification process for performing a detailed check on the model-to-scene correspondences is developed. Detailed experimental results are performed to confirm the feasibility and robustness of the recognition system.

An intuitive mathematical formulation is proposed for the interpretation of the geometric relationships of a triple of spatial edges and their perspective projection forming image lines. No restriction is imposed on the configuration of the triple of spatial edges. An eighth-degree polynomial equation explicitly defined by the space angles between the corresponding three spatial edges measured with respect to an object centered coordinate system is derived. The crux of this representation is that the angular attributes of pairs of spatial edges are object-independent. An effective hypothesis generation scheme is proposed which can take advantage of the commonality of this novel representation. It avoids replicating the same recognition module for every occurrence of the same triple feature in the same generic triple group. The groups are distinguished by the angles between the constituent model edges and do not involve any length metric property. Generally, a relatively small number of defined generic triple groups are employed to describe a wide range of polyhedral object models. Particular closed form solutions are derived for specific but common configurations of edges such as rectangular bar end and orthogonal triple. The practical significance and generality of our result are multifold. Extensive experiments are performed to verify the plausibility of employing connected triple edges and trihedral vertices as key features in the paradigm of hypothesis-generation and Hough-clustering approaches to object recognition. It is demonstrated that the accuracy of the estimated pose of objects is adequate.

Finally, outstanding problems identified and possible solutions to these problems are discussed. Future research directions are proposed.
## Contents

### List of Figures

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Introduction</td>
<td>vii</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Object Representation</td>
<td>1</td>
</tr>
<tr>
<td>1.3 Matching</td>
<td>2</td>
</tr>
<tr>
<td>1.4 Aim of Research</td>
<td>4</td>
</tr>
<tr>
<td>1.5 Summary of Contributions</td>
<td>7</td>
</tr>
<tr>
<td>1.6 Organization of the Dissertation</td>
<td>8</td>
</tr>
<tr>
<td>2 Survey</td>
<td>10</td>
</tr>
<tr>
<td>2.1 Object Representation</td>
<td>10</td>
</tr>
<tr>
<td>2.1.1 Criteria for Geometric Representation</td>
<td>11</td>
</tr>
<tr>
<td>2.1.2 Geometric Primitives</td>
<td>12</td>
</tr>
<tr>
<td>2.1.3 Representation Schemes</td>
<td>13</td>
</tr>
<tr>
<td>2.1.4 Discussion</td>
<td>14</td>
</tr>
<tr>
<td>2.2 Geometric Matching</td>
<td>14</td>
</tr>
<tr>
<td>2.2.1 Criteria for model invocation and hypothesis generation cues</td>
<td>15</td>
</tr>
<tr>
<td>2.2.2 Object Recognition Systems</td>
<td>16</td>
</tr>
<tr>
<td>2.2.3 Discussion</td>
<td>17</td>
</tr>
<tr>
<td>3 Analysis of Straight Homogeneous Generalized Cylinders Under Perspective Projection</td>
<td>19</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>19</td>
</tr>
<tr>
<td>3.2 A basic definition of an SHGC</td>
<td>21</td>
</tr>
<tr>
<td>3.3 Contour generators and their projection</td>
<td>22</td>
</tr>
<tr>
<td>3.3.1 Contour generators</td>
<td>22</td>
</tr>
<tr>
<td>3.3.2 Projection of contour generators</td>
<td>24</td>
</tr>
<tr>
<td>3.4 Analysis of circular SHGC's or solids of revolution</td>
<td>26</td>
</tr>
<tr>
<td>3.4.1 Symmetry of Projections</td>
<td>27</td>
</tr>
<tr>
<td>3.4.2 CSHGC's with Constant or Linear Sweeping Function</td>
<td>28</td>
</tr>
</tbody>
</table>
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1-1</td>
<td>An overview of the signal-to-symbol paradigm.</td>
<td>3</td>
</tr>
<tr>
<td>Figure 1-2</td>
<td>2D and 3D object representations</td>
<td>5</td>
</tr>
<tr>
<td>Figure 3-1</td>
<td>(a) A perspective projection of an SHGC. (b) A viewing coordinate system.</td>
<td>34</td>
</tr>
<tr>
<td>Figure 3-2</td>
<td>The projected contours of a CSHGC with an arbitrary sweeping function are symmetrical about its projected axis.</td>
<td>34</td>
</tr>
<tr>
<td>Figure 3-3</td>
<td>The contour generators of a right or oblique CSHGC with a constant sweeping rule are a pair of parallel straight lines.</td>
<td>35</td>
</tr>
<tr>
<td>Figure 3-4</td>
<td>The contour generators of a right or oblique CSHGC with a linear sweeping function are a pair of 3D convergent lines.</td>
<td>35</td>
</tr>
<tr>
<td>Figure 3-5</td>
<td>The contour generator of a right SHGC with an arbitrary and a constant or a linear sweeping rule is a list of straight lines.</td>
<td>36</td>
</tr>
<tr>
<td>Figure 3-6</td>
<td>The contour generator of an oblique SHGC with a constant or a linear sweeping rule is a list of straight lines.</td>
<td>36</td>
</tr>
<tr>
<td>Figure 3-7</td>
<td>Examples of object models</td>
<td>37</td>
</tr>
<tr>
<td>Figure 3-8</td>
<td>A simple example of decomposed subparts model invocation</td>
<td>37</td>
</tr>
<tr>
<td>Figure 4-1</td>
<td>An example of junction configurations</td>
<td>40</td>
</tr>
<tr>
<td>Figure 4-2</td>
<td>Possible interpretations of triples</td>
<td>41</td>
</tr>
<tr>
<td>Figure 4-3</td>
<td>A closed chain test is performed on a processing chain</td>
<td>42</td>
</tr>
<tr>
<td>Figure 4-4</td>
<td>(a) An unique single point merging triple (b) An unique single segment merging triple (c) An unique double segment merging triple.</td>
<td>43</td>
</tr>
<tr>
<td>Figure 4-5</td>
<td>Examples of feasible open polygons</td>
<td>44</td>
</tr>
<tr>
<td>Figure 4-6</td>
<td>The grouping results</td>
<td>46</td>
</tr>
<tr>
<td>Figure 5-1</td>
<td>See text</td>
<td>65</td>
</tr>
<tr>
<td>Figure 5-2</td>
<td>Triangle-pair features</td>
<td>66</td>
</tr>
<tr>
<td>Figure 5-3</td>
<td>Object Models</td>
<td>66</td>
</tr>
<tr>
<td>Figure 5-4</td>
<td>The detailed matching results</td>
<td>67</td>
</tr>
<tr>
<td>Figure 5-5</td>
<td>The scene image and the experimental results</td>
<td>68</td>
</tr>
<tr>
<td>Figure 5-6</td>
<td>The detailed matching results</td>
<td>69</td>
</tr>
<tr>
<td>Figure 5-7</td>
<td>The detailed matching results</td>
<td>69</td>
</tr>
<tr>
<td>Figure 5-8</td>
<td>The scene image and the experimental results</td>
<td>70</td>
</tr>
<tr>
<td>Figure 5-9</td>
<td>The scene image and the experimental results</td>
<td>71</td>
</tr>
<tr>
<td>Figure 6-1</td>
<td>Analysis of a spatial edge in general and canonical position</td>
<td>79</td>
</tr>
<tr>
<td>Figure 6-2</td>
<td>An analysis of a triple of spatial edges in the camera $F_C$, canonical $F_A$ and $F_B$ frame</td>
<td>80</td>
</tr>
<tr>
<td>Figure 6-3</td>
<td>see text</td>
<td>93</td>
</tr>
<tr>
<td>Figure 6-4</td>
<td>Computation of a translation vector in the general triplet case</td>
<td>97</td>
</tr>
<tr>
<td>Figure 6-5</td>
<td>see text</td>
<td>101</td>
</tr>
<tr>
<td>Figure 6-6</td>
<td>The vertices of a model</td>
<td>108</td>
</tr>
<tr>
<td>Figure 6-7</td>
<td>Geometric Constraint Equations</td>
<td>109</td>
</tr>
<tr>
<td>Figure 6-8</td>
<td>The constraint angle solutions and estimated object’s poses</td>
<td>109</td>
</tr>
<tr>
<td>Figure 6-9</td>
<td>The experimental results from synthetic test image</td>
<td>110</td>
</tr>
<tr>
<td>Figure 6-10</td>
<td>Perturbation: $\triangle : 0.5 + : 1.0 \times : 1.5 \cdot : 2.0 \diamond : 2.5 \ast : 3.0$</td>
<td>112</td>
</tr>
<tr>
<td>Figure 6-11</td>
<td>A hexagonal polyhedron model and a pyramid model</td>
<td>115</td>
</tr>
<tr>
<td>Figure 6-12</td>
<td>Extracted image lines and their corresponding model edges</td>
<td>115</td>
</tr>
<tr>
<td>Figure 6-13</td>
<td>Geometric Constraint Equations</td>
<td>116</td>
</tr>
<tr>
<td>Figure 6-14</td>
<td>The constraint angle solutions and estimated object’s poses</td>
<td>116</td>
</tr>
<tr>
<td>Figure 6-15</td>
<td>The experimental results using a real image</td>
<td>117</td>
</tr>
<tr>
<td>Figure 6-16</td>
<td>The best hypotheses and the final match</td>
<td>118</td>
</tr>
<tr>
<td>Figure 6-17</td>
<td>The image connected triples and the number of constraint angle sets</td>
<td>121</td>
</tr>
<tr>
<td>Figure 6-18</td>
<td>The image trihedral vertices and the number of constraint angle sets</td>
<td>121</td>
</tr>
<tr>
<td>Figure 6-19</td>
<td>The description of widget model</td>
<td>122</td>
</tr>
<tr>
<td>Figure 6-20</td>
<td>Representative Hypotheses</td>
<td>123</td>
</tr>
<tr>
<td>Figure 6-21</td>
<td>Representative Hypotheses</td>
<td>124</td>
</tr>
<tr>
<td>Figure 6-22</td>
<td>Representative hypotheses, best hypotheses and matching result</td>
<td>125</td>
</tr>
<tr>
<td>Figure 6-23</td>
<td>The descriptions of the HEXM, house and box models</td>
<td>128</td>
</tr>
<tr>
<td>Figure 6-24</td>
<td>The descriptions of generic triple groups of the 3 models</td>
<td>128</td>
</tr>
<tr>
<td>Figure 6-25</td>
<td>The extracted scene image features and the constraint angle solutions</td>
<td>129</td>
</tr>
<tr>
<td>Figure 6-26</td>
<td>A test scene image and infeasible hypotheses</td>
<td>130</td>
</tr>
<tr>
<td>Figure 6-27</td>
<td>Infeasible hypotheses</td>
<td>131</td>
</tr>
<tr>
<td>Figure 6-28</td>
<td>The best hypotheses and the correct matches</td>
<td>132</td>
</tr>
<tr>
<td>Figure 6-29</td>
<td>The matching results from the real image involving multiple target objects</td>
<td>133</td>
</tr>
</tbody>
</table>
Glossary

(a) **Junction**: is defined by two straight line segments whose endpoints are mutually proximal to within a prespecified threshold, measured along the line of intersection.

(b) **Triple**: is formed by combining pairs of junctions which share a common straight line and their resultant configuration must have only two endpoints.

(c) **Trihedral Vertex**: is a 3D spatial vertex with three space lines emanating from a common tip.

(d) **Generic Rectangular Bar End**: is defined by three space lines where one of the edge directions is orthogonal to the other two space lines.

(e) A chain or polygonal curve is defined by a list of connected straight line segments. An **open polygon** is defined by a chain whose endpoints are isolated. A **closed polygon** is defined by a chain whose endpoints are connected.
Chapter 1

Introduction

1.1 Introduction

During the past ten years, the application of computer vision systems has become increasingly important and widespread. Areas of application include robotics, remote sensing, industrial inspection and automated guidance and tracking of objects. In particular, automatic visual inspection and robotic planning are two fields which strongly motivate the research of computer vision. Many industrial inspection and assembly tasks require a degree of visual capability. Although, human vision system is very flexible its performance is variable, depending on several factors such as motivation, fitness, experience and the conditions of the working environment. Generally, the efficiency of human vision will be reduced due to fatigue and lack of motivation. In contrast machine vision is not affected by these factors and can in principle be much more consistent over long periods of time without degrading its performance.

Many manufacturing systems use robots in repetitive operations such as lathe turning, metal cutting, milling, hole drilling and bolt insertion, car washing etc. Some robotic machines could be used to perform their task in hazardous areas such as nuclear plant, high-voltage room and vacuum space. Most of these existing robotic systems do not receive any feedback from the working environment. As a result, the movement of the robotic manipulator can only operate in a restricted area and they cannot respond to any error or correct for any misplacement of the feed components. For example, a car washing robot may continue to spray water from a nozzle even if a car body is not present in the working area. This causes a waste of energy resources. The application of vision to these areas would provide visual feedback to remedy such problems. Moreover, it would offer new opportunities for the development of flexible automation systems which do not require costly modification of hardware between product changes, leading to lower production cost, increased reliability, accuracy and serviceability. The early failure could be detected before severe damage occurs.

In the absence of 3D laser range finders, which in any case are either too expensive, provide a slow 3D depth data acquisition or are difficult to calibrate, the recognition task may be based on a
single gray-level image acquired under perspective projection. Image interpretation based on multiple images and range data is outside the scope of this thesis.

The goal of computer vision is to construct a scene description based on the information extracted from an image or image sequence. A general model-based vision system operation can be decomposed into four levels of processing. An overview of these processes is given in Figure 1-1. These processes can be summarised as follow:

- **Pre-processing**: The fundamental requirement for computer processing of image data is that the images must be available in a digital form. The pre-processing is the conversion of analog sensory data to a digital form. The digital form is an array of finite length binary words which is suitable for further processing.

- **Low-level processes**: From the information content point of view image data is degraded as a result of shadows, complicated lighting environment and transmission noise. Image enhancement may be used to accentuate certain useful image features for subsequent image analysis. The low-level processes manipulate pixel information of digital images and extract certain features that are more manageable and useful for identification of objects. These processes in the analysis chain produce low-level symbolic events such as regions with uniform characteristic and edge points with their attributes. These features are labelled and attributed with their property value.

- **Intermediate-level processes**: In this processes, low-level symbolic events are grouped and reorganised according to their characteristics and geometrical properties. Examples of intermediate events are parallel and collinear line features, circles, ellipses, general curves and ribbons. These events can be represented by a relational structure such as a graph whose nodes represent regions, labelled with various quantitative attributes such as shape, color, texture etc, and arcs which represent relationships between nodes.

- **High-level processes**: These processes use the image description given in terms of an attributed relational graph produced by intermediate-level processes to assign object names to groupings of scene feature. Search strategies are used to find the correspondence between the intermediate-level descriptions of a scene and knowledge of stored object model.

The above mentioned processes are essential to deriving the interpretation of pixels in terms of the names of objects and their relationships. Moving from pixels to objects is an example of the signals-to-symbols paradigm.

### 1.2 Object Representation

In a general vision system, multiple representations are essential for a succinct signal-to-symbols conversion. The processes start from pixel-level signals and describe successively more organised
Figure 1-1: An overview of the signal-to-symbol paradigm.
and goal-dependent data groupings. At each level of processing, the definition of a natural vocabulary is required in order to make desired information explicit.

Both computer graphics and vision representations need similar geometrical features, but the storage, utilization and level of detail of spatial information are quite different. In computer graphics, visual realism of an object model is paramount. Object representation for computer graphics does not generally adequately represent functional and conceptual structure so that it is amenable for manipulation by a machine-reasoning technique. In computer vision, the choice of object representation is of crucial importance. A good representation scheme should be able to facilitate the implementation of algorithms for image interpretation.

Image scene features extracted by the grouping processes also have to be represented in a form which is suitable for use with matching algorithms. Generally speaking, objects can be described by 2D or 3D primitive features.

Figure 1-2 (a) shows how the image of a cup might appear as a result of occlusion, complicated lighting, mutual illumination, etc. In this image, 2D low-level geometric features such as circular arcs, curves and linear segments may be extracted. These features are viewpoint dependent and may be inaccurate and incomplete. The image description is two-dimensional while the stored object models are three-dimensional. Figure 1-2 (b) shows the object model is represented by a cylinder and part of a torus. The goal of image interpretation is to relate the image and model representation in such a way that a scene description can be concluded. The implementation of the algorithms for image interpretation is greatly facilitated and simplified if the same representation for both the scene image and object model is used. Generally, this may not be possible due to constraints such as computational load and working environment etc. In that case the image and model representation schemes should be implemented in such a way that the transformation from one representation to the other can be easily computed. An immediate question is which is the best object representation scheme that should be implemented and used to facilitate the geometric matching process. What are the criteria for selecting these representation schemes. These questions are addressed in the next chapter.

1.3 Matching

Some of the earlier research in low-level and intermediate-level processes concentrated on the extraction of symbolic features such as lines, arc etc. These features are individually not too informative to draw inferences concerning the presence of an object. A bottle can't be recognised by only a straight line. These features are used in a "bottom-up" approach in which models are defined in terms of constraint measures on these features. Image interpretation is accomplished only when many low-level features acquired from the scene are consistent with the low-level description of object models. General "rules of form" can be used to initiate "bottom-up" grouping operations that organise low level data into larger scale structures but they are not powerful enough to resolve all...
Introduction

Figure 1-2: 2D and 3D object representations

the ambiguities found in poor quality image descriptions. In a simple scene, this approach may be computationally feasible, but the complexity of the real-world imposes the need for a more reliable and effective matching paradigms. Furthermore, it is not trivial to control the growth and complexity of the potential number of 2D model-feature correspondences.

At the other end of the spectrum, model-based vision concerns the use of high level models to guide image interpretation. Methods of object recognition which exploit prior knowledge of object models for organizing the processing are known as “top down” approaches. The idea behind the approach is that high level, object specific models can provide much stronger predictions for the object matching process. This “top-down” flow of information can be used to improve both the robustness and efficiency of the matching process. Geometric models, in particular, can provide highly constraining predictions for image-object matching.

To date, many existing systems exploit a combination of the “bottom-up” and “top down” approach to tackle the inherently difficult recognition problem. This approach first finds relevant low-level data features then significant features are used to identify low-level feature combinations that satisfy the description of features of object models. Plausible object models are chosen from the knowledge-base for quantitative matching; this process is known as hypothesis generation. Further search is used to verify the presence of predicted additional features describing the hypothesized object. This approach substantially depends on pre-computation of salient and distinct model features such as trihedral vertices, connected edges, parallel arc etc, and cliques of such features.
The main aim of the above strategies is to find the correspondence between the representations of the object and image data, and the pose of the target object in the camera coordinate system. The goal of image interpretation is difficult for several reasons:

- An image description is usually imperfect in many ways. Particular effects include:
  - Extraneous data: an image usually contains data from many objects as well as spurious data caused by imaging and poor segmentation processes.
  - Missing data: data corresponding to a particular object is almost always missing. This can be caused by physical effects such as occlusion or may again be the result of poor segmentation processes.
  - Inaccurate data: image discretisation effects and poor segmentation mean that estimates of parameters which characterise an object are always inaccurate.

- As the number of scene objects and possible models grows the number of potential matches grows exponentially. This is mainly because the currently used representation of the models are not generic.

- Primitive features of higher-order structure provide tighter constraints and fewer ambiguities. However, the analysis of such features under perspective is inherently difficult. The resultant formulations usually do not give much insight into the nature of the problem. They are also less numerous and more difficult to get from images.

- It is not obvious how to establish model and scene feature correspondences from a cluttered environment. This is more difficult when dealing with matching of multiple scene objects to multiple object models. A major problem is the inability to control the growth and complexity of the assignments of corresponding feasible token features.

Some of the above problems are also encountered in recognition systems based on other approaches such as stereo matching and range image understanding. Many existing recognition systems based on a single perspective image have tried to solve the above mentioned problems partially or completely. Several distinct phases can be usefully distinguished in their operation. There are two off-line stages:

- model generation To construct a database of models;
- model analysis To identify and organise model features into structures for matching and to develop strategies for execution of the matching task;

and several run-time stages:

- model invocation To identify in the object database suitable candidates which are consistent with observed image descriptions. At this level, qualitative image descriptions are generally
used for initializing the search for subsets of feasible geometric features of the models. For example, an image trihedral vertex is a clue for spatial corners of object models.

- hypothesis generation In this phase, quantitative metric information of the models and corresponding image features are employed to derive the geometric relationships between model and scene features. The geometric feature assignments are then checked using the geometric constraints derived from the model and scene correspondences. The translation and rotation transforms of the plausible models associated with geometrically admissible solutions are computed.

- model verification The hypothesized objects are backprojected onto a 2D image plane. The predicted 2D features are used to perform a detailed check between the predicted model features and feasible features extracted from the scene image, confirming features present and accounting for features which are not observed.

- consistency verification To check spatio/temporal coherence of the interpretation.

In some of the recognition systems, the modules of model invocation and hypothesis generation are embedded into one module. Some of the above mentioned problems are also encountered in recognition system based on other approaches such as stereo matching and range image understanding. Two limitations of the existing methods are that systems are very sensitive to object occlusion and the efficiency of the system decreases substantially when the knowledge-base of the object models is very large.

1.4 Aim of Research

Representation is a key and all pervasive issue in computer vision. The solution of many vision problems can be greatly simplified by correct choice of representation. One of the purposes of this dissertation is to develop a modelling system for describing both curved and polyhedral objects.

Many man-made objects are polyhedrons or partially polyhedral in shape, especially industrial components, domestic utensils, furniture and buildings, etc. Hence, the problems of recognizing polyhedral objects from a single perspective image is a practically important problem in computer vision. Based on this approach, there are still many unsolved or partially solved problems. The key purpose of the research presented in this dissertation is to provide solutions to some of these problems.

1.5 Summary of Contributions

- A mathematical framework for analysing SHGC's under perspective has been developed. Using the framework a number of results concerning geometric properties of the contour generators
Introduction

of straight homogeneous generalized cylinders (SHGC's) have been derived and discussed. The 2D projected contours are analysed under perspective projection. The right and oblique SHGCs with circular and arbitrary cross-section are considered. An analytical expression for the contour generators of SHGCs and their projections under perspective, with the camera pointing at an arbitrary point of interest has been derived.

- A novel algorithm for the detection of polygonal curves has been developed. The algorithm exploits specific heuristics which greatly reduce the combinatorial explosion associated with unconstrained combination of linear segments into higher level polygonal structures. The feature finder is useful for both object recognition and pose determination.

- A hypothesize-verify paradigm based on local shape descriptions, namely triangle-pair and quadrilateral features, has been developed. Two geometric constraints, namely distance and angle constraints, are derived and incorporated in the matching process to limit the size of the search space. Many implausible hypotheses generated from scene-model triangle-pairings and quadrilateral feature assignments are pruned away at the early stage of the processing stage by using geometric constraints. The most plausible hypotheses are searched for by scanning through a one dimensional confidence measure plot.

- An intuitive mathematical formulation has been proposed for the interpretation of the geometric relationships of a triple of spatial edges and their perspective projection forming image lines. No restriction is imposed on the configuration of the triple of spatial edges. An eighth-degree polynomial equation explicitly defined by the space angles between the corresponding three spatial edges measured with respect to an object centered coordinate system is derived. The crux of this representation is that the angular attributes of pairs of spatial edges are object-independent. This angular representation has important implications on model invocation and matching efficiency. An effective hypothesis generation scheme has been proposed which can take advantage of the commonality of this novel representation. Particular closed form solutions have been derived for specific but common configurations of edges such as rectangular bar ends and orthogonal trihedral vertex. A polyhedral object recognition system based on triples of connected edges has been developed. The practical significance and generality of our result are multifold.

- Extensive experiments have been performed to verify the proposed methods for solving the identification and pose estimation problem. The effectiveness, reliability and capability of our system have been verified in the domain of polyhedral world.

1.6 Organization of the Dissertation

The dissertation is organized as follows.
In Chapter 2, a survey on object representation and recognition systems is presented.

In Chapter 3, a mathematical framework for analysing SHGC's under perspective projection is described. A number of results concerning geometric properties of the contour generators of straight homogeneous generalized cylinders (SHGC's) are derived.

In Chapter 4, an algorithm for the extraction of triple and polygonal curves is presented. Specific heuristics rules for reducing the combinatorial explosion of the grouping process are mentioned.

Chapter 5 presents a hypothesize-verify paradigm based on local shape properties, namely triple-pairing and quadrilateral features. The analysis of two geometric constraints, namely distance and angle constraints, is described. The integration of features into the matching process is described. A method of pose estimation is described. The matching strategies of the recognition system are presented. Finally, the details of the experimental results are presented.

Chapter 6 presents an intuitive mathematical formulation for the interpretation of the geometric relationships of a triple of spatial lines and their perspective projection forming image lines. An effective hypothesis generation scheme is described. The integration of model organization, invocation, hypothesis and the verification process for recognizing polyhedral objects is presented. Extensive experiment results are reported.

In Chapter 7, the research work is summarized. Problems associated with the work are described. Finally, possible future research work is described.
Chapter 2

Survey

Representation is a key and all pervasive issue in computer science. The solution of many problems can be greatly simplified by correct choice of representation. The method of object representation is highly influential on the choice of matching process.

A comprehensive literature survey has been carried out to establish the current state of the art. An attempt has been made to identify the commonality of approaches and the weakness of current methodology. Issues which have been of particular concern are robustness and accuracy of methods, flexibility of the approaches and computational feasibility and efficiency. In this chapter, a summary of the survey on object representation and recognition systems is presented.

2.1 Object Representation

In this section, a summary of the survey on object representation is presented. For a more detailed version of our survey the reader is referred to (Wong [72]). Other comprehensive and critical literature survey and overviews include (Ballard [3], Besl [8], Chin [21]).

Both computer graphics and vision representations need similar geometrical features, but the storage, utilization and level of detail of the spatial information are quite different. In computer graphics, visual realism of an object model is paramount. In computer vision, the expressive power of the geometrical information of an object model is of crucial importance. The method of object representation is highly influential on the choice of matching process.

In Section 2.1.2, different types of geometric primitives exploited in existing recognition systems are described. In Section 2.1.3, a summary of the survey on different shape representation schemes relevant to the object recognition task is presented. In the next section, an outline of the desirable properties of a geometric representation will be presented.
2.1.1 Criteria for Geometric Representation

In this section we discuss some of the desirable criteria which have to be borne in mind when choosing a geometric representation for matching. Several authors have addressed these issues including Marr [52], Brady [15], Biederman [9] and Leavers [47]. These can be described as follows:

- **Generality**: what is the range of geometric objects which are representable and what fraction of real-world objects do they include? For example, it is often stated that somewhere between 80-90% of man-made industrial objects can be well described by linear and quadratic forms and thus this is a reasonably good way to represent this limited domain. However, far fewer non-rigid, natural forms can be represented by linear and quadratic forms.

- **Naturalness**: this relates to the direct correspondence between the representation and physical form in the world. For example, a circular curve can be represented to an arbitrary accuracy by a large number of very small linear segments. However it can be represented much more naturally as a circle by specifying the parameters which denote the center coordinates and the radius. More natural representations are normally more concise and afford more efficient computation.

- **Computability**: it is necessary to demonstrate that any proposed representation of shape is computationally feasible. This computational feasibility relates to two aspects. Firstly, it must be shown that descriptions can be reliably extracted from image data. Secondly, tools and techniques must exist to construct the models and permit relevant predictions of image appearance to be made [52].

- **Decomposability**: the development of larger structures from smaller ones is a powerful principle. Thus a good representation should contain primitive or irreducible components which can be combined in many different ways. This is the concept of developing a hierarchical geometric description based around part and sub-part relationships [52], [9].

- **Spatial Relations**: it is not sufficient to decompose the image into its constituent shape primitives. In addition, the representation should also make explicit the geometric and spatial relations between those shape primitives [52], [47].

- **Stability**: the representation should not depend too strongly on absolute judgements of quantitative detail. It should be robust, especially at low levels, to the occurrence of moderate amounts of noise. An example of a representation which does not meet this criteria is the medial axis transform of a shape. A small, possibly noise induced, perturbation to the boundary contour of a shape can cause a large change in the resulting medial axis description [52], [47].

- **Similarity**: to be useful for recognition, the similarity of two shapes should be reflected in their descriptions but at the same time the representation should also encode subtle differences [52], [15].
Survey

- **Saliency**: the representation should be such that gross or salient features associated with shape can be made explicit and used to bring essential information to the foreground allowing smaller and more easily manipulated descriptions to suffice. This corresponds to the idea that not all pieces of information are equally informative and a key idea is to exploit this principle to achieve efficient interpretation. This means that models should be hierarchical and adapted to the specific goal to be achieved [9].

- **Invariance**: information that is the basis of matching should be invariant to operations such as translation, rotation, scale and projection. The consequence of this principle is that all views of an object could be directly matched to a single model. In practice, this is often only partially achieved [52], [15], [9].

- **Extensibility**: in order to cope with large numbers of models a good representation should be extensible to include new models with a sub-linear cost. In most representations individual object models are totally independent. Thus, features will be repeated if they appear in more than one model. A good representation should support the ability for common features to be represented only once but shared by several models. This means that a good representation should incorporate a hierarchy of features [52].

- **Local support**: in a complex scene occlusion is a very serious problem. It is therefore desirable that the representation adopted for matching has local support i.e. it can be computed from purely local measurements [52], [15].

- **Tolerance**: as noise and uncertainty are omnipresent in image data it is necessary to have ways to take them into account. Thus a good representation must be able to describe tolerances on parameters and spatial dimensions.

The above mentioned criteria will be employed as a reference to select geometric feature primitives for using as key or seed features for model invocation and hypothesis generation.

2.1.2 Geometric Primitives

**Points, Lines and Curves**

To date most work on object matching has used point or curve data. There are several reasons for this. One reason is the relative ease and reliability of extraction of these primitives. Another reason is that the projected extremal boundaries of objects directly yield image curves. Similarly points of high curvature have a fairly direct interpretation in terms of physical scene events. Hence, a lot of the physical structure of a scene can be recovered from point and curve data. The rapid interpretation of silhouettes suggests that curve data contains many of the most salient features used in object recognition. It is also important to note that most established geometric matching techniques are based around point (Fischler [28]) line, (Lowe [50], Horaud [35]) and curve (Koch [45], Chien
Survey

[20] primitives. The major problems associated with points, lines and curves for matching are the abundance of this type of primitive and their production as artifacts of the imaging and segmentation processes. Many effects other than extremal object boundaries can give rise to gray level curves i.e. changes in surface orientation, changes in surface properties and shadowing effects. This extraneous curve data adds to the combinatorics of the matching effort. Another common problem is that some processes such as curve extraction are often imperfect and produce fragmented data. This implies the development of extra grouping processes, possibly model driven, to recover faithful data for matching. Finally, the exact way in which curve data is used for recognition is open to question. As well as recognising silhouettes humans are able to easily interpret caricatures or cartoons which are not faithful geometric representations of the objects which they portray.

Surfaces and Volumes

Representative works based on surface representation for object recognition are (Oshima [55], Grimson [29], Fan [27]). Recognition systems based on volumetric description are (Binford [10], Brooks [16], Pentland [56]). It is much less well established than points or contours although they provide a much more natural and compact description of many objects. The number of surfaces which compose an object is generally quite small and therefore the scale of the matching problem using surfaces is intrinsically small. The spatial relationship between surfaces is also more constraining than that between curves and this rich structural information can be used to efficiently reduce ambiguity of image interpretation. The major difficulty with the use of surface information is that specialist sensors are generally needed to derive it reliably. There are not many good, practical procedures for the extraction of surface information from gray level imagery. Methods based on disparity measurements, either by multi-camera systems or the motion of a single sensor, only provide depth estimates at places of correspondence such as along edges or at vertices. Interpolation from these sparse points is notoriously difficult, especially in the presence of discontinuities. An interesting alternative is to infer qualitative surface shape information from the sparse depth estimates.

2.1.3 Representation Schemes

A survey on different shape representation schemes relevant to the object recognition task have been carried out. In here we shall summarize the results. Methods used to describe 3D objects can be categorized into three different major groups:

- **2D representations**: Some examples are B-spline (deBoor [14]) and Codon primitives (Richards [62]).
- **2½D representations**: Some examples are Relational Surface Patch (Oshima [55]), Characteristic View (Koenderink [46], Chakravarty [18]) and Extented Gaussian Image (EGI) (Hom [39]).
• 3D representations: Some examples are Wire-frame (Wesley [70]), Surface Boundary (Baumgart [7]), Constructive Solid Geometry (CSG) (Voelcker [67]), Cell decomposition (Jackins [41]), generalized cylinder (Binford [10], Shafer [63], Ponce [58]) and Superquadric primitives (Pentland [56]).

Some researchers (De Floriani [23], Hansen [30]) have already studied the use of explicit 3D geometric models for object recognition. These models have been frequently constructed using standard CAD tools. The advantages of using CAD modellers are: they are tools which already exist; they describe 3D objects unambiguously; useful data can in principle be computed from a model object for matching purposes; the description used is rich and compact. The disadvantages are: the underlying data structure of an object model is very complicated; extraction of geometrical information is computationally expensive; some information may not be available or organised in natural structures.

2.1.4 Discussion

A wide range of techniques has been reported suitable for real world objects for accomplishing the recognition task. However, there is no single effective representation scheme for describing general shape.

In order to use the commercial CAD solid modeller in a recognition system, proper transformational procedures need to be developed to transform the description of a CAD model to a useful geometrical information for matching purpose.

A wireframe [70] or a boundary representation [7] scheme is feasible and more natural for describing models within the domain of polyhedral objects. The transformations from the standard surface-edge-vertex data structure to key features employed in the current existing recognition systems are computationally expensive but simple in implementation. Moreover, the transformation procedures involved in the model organization module can be performed off-line.

In the case of modelling 3D curve objects, it is more suitable and compact to model them by using volumetric representation such as cell decomposition representation [41], generalized cone [10] [58], and superquadrics representations [56]. The complex shapes are more naturally described by decomposition into a number of simpler volumetric primitives modelled by one of the above representation schemes. However, the stability of the decomposition process is still an open issue for further research.

2.2 Geometric Matching

In this section, we consider the issues which concern model invocation, and hypothesis generation. We also analyse the criteria for effective key or seed features to initial the matching search space. Several established and more recent model-based recognition methods have been reviewed. The main
goal of these recognition systems is to determine the identity and orientation of the object which is currently being viewed by comparing an explicit geometric description of the object model to image data.

Established techniques employed in model-based vision rely largely on geometric and spectral cues. More recent methods have begun to address larger scale databases of models and the use of higher level world and scene knowledge.

In here, a summary of the survey will be presented. For more details, the reader is referred to (Wong [72]). Other comprehensive and critical literature survey and overviews include (Binford [11], Besl [8], Chin [21]).

2.2.1 Criteria for model invocation and hypothesis generation cues

Any object property is potentially useful for model invocation. However not all pieces of knowledge or evidence are equally good. Some of the criteria which must be considered when choosing pieces of evidence or 'cues' for model invocation and hypothesis generation are:

- **Distinctiveness**: this is an evident requirement for a useful cue as model invocation concerns discrimination among objects.

- **Saliency**: this refers to the prominence, conspicuousness or noticeability of an object property.

- **Computational cost**: as with all vision processes this should be kept as low as possible.

- **Robustness**: if model invocation is to be successful then it must be able to cope with poor quality data.

The above attributes represent distinct concepts but all are not necessarily mutually independent. In the context of model invocation, distinctiveness and saliency are often confused and the terms are sometimes used interchangeably. Both are important in model invocation. Distinctiveness refers to the uniqueness of the feature and therefore the potential power that its observation has in constraining the number of models that are invoked. However, two similar models may only be distinguishable by observation of some small feature which is extremely difficult to find, especially in poor quality data. In such a case the small feature is distinctive but of little use as a cue. Saliency refers to the prominence of a feature. Highly salient features should be rapidly detectable and thereby provide efficient and robust information for model invocation and hypothesis generation.

Computational cost is obviously something which must be low for an effective cue. The purpose of a cue is to "suggest" highly likely models for further study and therefore rapidly prune the number of models under consideration.

The unreliability of image description processes mean that cues need to be chosen so that they are tolerant to image degradation. Cues which are based on strict quantitative measures are likely to fail for a variety of reasons associated with both poor image segmentation processes and physical
effects such as object occlusion. It is therefore better to base cues on qualitative image features such as structural relationships between image primitives. An alternative way to enforce some degree of robustness in model invocation is to exploit redundancy and base model invocation around the observation of one or more members of a small subset of independent cues. This also has the desirable property that finer distinctions are possible between models.

An issue which is of particular concern when matching 3D models to 2D image data is the variation of the object feature appearance as a function of viewing position. In the model invocation stage of object matching object pose is unknown and exhaustive matching to all possible 2D projections is computationally too expensive. Hence if 2D descriptions are to be used it is necessary to find ways of inferring model identity which are insensitive to viewpoint.

2.2.2 Object Recognition Systems

In this section, the development of ideas relating to well established methods of geometric cueing are described. Typically these methods rely on geometric cues. Several examplar systems are discussed. This is then followed by a review of more recent work which is less well tested but is beginning to show what implications there are in moving from systems with few models to more realistic large size systems. In here, we shall give a summary of our review. For more details, the reader can be referred to [72].

One of the earliest, successful areas of computer vision was industrial inspection and part handling which involved binary images of isolated, rigid, flat objects. In these particularly simple cases model invocation was achieved using global object properties, such as object area or rough shape measures based on moments. However, later work considered more complicated situations where objects were allowed to overlap. In these cases global measures were not useful and were therefore superceded by methods based on local shape properties of objects.

The ACRONYM [16] vision system clearly demonstrated the potential power of geometric models to constrain 3D object recognition and was influential in coining the term model-based vision. Around the same time other work was being pursued on the exploitation of geometry. Bolles and Cain [12] developed the local, focus feature idea whereby a group of features could be used as a cue for indexing into a 2D model. Lowe[50] experimented with groups of features in his SCERPO system and utilised the idea of perceptual groupings i.e. non-accidental structures which possess a degree of viewpoint invariance. Many systems were built around variants of these ideas using other types of compound primitives and other types of data such as stereo [59] or depth data [33], [13].

An important idea dating back to the work of Marr and Nishihara [51] is that of object representation by component parts and invoking models by parts recognition. Brooks[16] exploited it in the ACRONYM system using ellipses and ribbons as geometric cues into generalised cylinder models of objects. It has received renewed interest more recently with the suggestion of superquadric models[56] or geons[9] as generic representational primitives. However, thus far none of these systems has been
demonstrated to work well on a realistic problem.

A lesson of early work in computer vision was that it is often impossible to make accurate and precise measurements. In addition, accurate world models are difficult to devise. Thus, a strong school of thought proposes that robust vision should be based on qualitative measurements. In relation to model matching this has led to the exploration of topology as a supplement to geometry for model invocation. In particular, an important development is the concept of equivalence of large sets of views of a 3D object. Thus objects can be recognised over a large range of views by comparison with a few prototypical or characteristic views [46] [18]. In practical systems the equivalence classes are determined by tesselating the viewsphere around an object, projecting the model onto each tessel. The features found are then analysed and all tessels with similar features are grouped together.

In most of the work cited above little formal account was taken of the saliency and distinctiveness of cues. Properties were given rankings and sought for in an order which reflected intuitive notions of their value. A typical heuristic was that used in the HYPER [2] recognition system where matches were only made with the 10 longest line segments of a model. Tumey, Mudge and Volz [66] were one of the first authors to formalise a notion of the saliency of cues and devise an algorithm for quantitative weighting of geometric features. Their analysis involved optimisation of a criterion function which measures the correlation between subparts of shape models. Subparts which were most unlike others are given a high weighting in the matching process. Wallace and McAndrew [68] have more recently suggested a probabilistic framework within which to consider the problem.

Examples of recent work (Lehrer [48], Hoffman [32] Burns [17], Hansen [30] ) in model invocation are building on previous studies but several distinct themes are beginning to emerge. There is a growing realisation that the extension to large model databases not only aggravates existing problems but also introduces new problems which do not have to be addressed in smaller scale problems. New techniques have to be devised for representing models, organising predictions and developing task-solving strategies. To achieve robust discrimination between many similar models knowledge and evidence from many different sources have to be integrated. A large part of many methods is a detailed analysis of the model database prior to run-time execution in order to perform and precompile model predictions.

2.2.3 Discussion

In this section, we have considered the literature addressing methodologies and paradigms used in existing recognition systems. The crucial and difficult tasks for general model-based recognition are: selecting feasible objects from large set of distinct objects and establishing their correspondences with the scene features. It is clear that model invocation and hypothesis generation from large databases is an area of topical interest and that this necessitates significant changes to the way that geometric and other information is modelled and organised. There is also much potential in the exploitation of high level knowledge and the application of more formal methods of evidence combination. However,
many of these methods have been experimented with in only very limited domains and their full exploitation is a long term development.

In this dissertation, we will address some of the difficulties encountered in recognizing polyhedral objects based on a single perspective image using a small size of model base. In spite of its limited horizons, it will be shown that the novel paradigm proposed can be easily modified and extended to deal with the recognition of many object models.
Chapter 3

Analysis of Straight Homogeneous Generalized Cylinders Under Perspective Projection

3.1 Introduction

Representing objects using generalized cones as 3D volume primitives was originally proposed by Binford [10], Agin and Binford [1]. A generalized cylinder (GC) is a solid volumetric component which is generated by sweeping a planar surface along its axis or spine. A rule for sweeping the planar surface is used to govern the shape and size of a generalized cylinder. The planar cross-section surface can have an arbitrary shape although a particularly simple and common class might be a circular, elliptical or polygonal cross-section. The axis or spine of a GC is not necessarily straight, it can be a space curve. The sweeping rule is not necessarily constant, it can be either linear or non-linear. The angle between the cross-sections and the axis must be constant but not necessarily orthogonal.

Many researchers have considered GCs as significant features for modelling and representing 3D objects in general vision systems. Nevatia and Binford [54] have developed a laser triangulation ranging scheme to obtain three-dimensional coordinates of all visible surface points from a single view. Occluding contours from range data have been used to reconstruct generalized cylinder descriptions. Scene analysis is performed by matching these reconstructed descriptions against descriptions of the stored models represented by GC primitives. The arbitrary planar cross-sections of GCs used in their work are normal to its axis. The center of gravity of each cross-section must pass through the axis. Marr and Nishihara [51] used 2D image data from a scene to recover the descriptions of GCs. They assumed that the contour generators of the observed scene objects viewed under orthographic projection are planar. The generalized cones used in their approach are plain cylinders each described by a circular cross-section, a straight axis and a constant sweeping rule. Brooks [16] designed and
developed a model-based image understanding system called ACRONYM. The system clearly demonstrated the potential power of geometric models represented by GC primitives to constrain 3D object recognition. The GCs used in ACRONYM system are characterized by polygonal and circular arc cross-sections. The sweeping-rule can be linear or non-linear for GCs with a straight axis and must be constant for GCs with a circular arc axis. Vision systems using GC primitives to describe 3D objects shall include large classes of GCs for modelling complex objects. Useful geometrical properties of large classes of GCs could be evaluated using the theory of differential geometry. Shafer and Kanade [63] established a rigorous mathematical basis for analysing GCs. Various classes of GCs have been categorized according to the taxonomy proposed and developed in their work. Shafer proved several interesting properties such as planarity etc. of straight homogeneous GC (SHGC).

Ponce and Chelberg [57] studied a larger class of GCs. The GCs discussed in their work are solids of revolution, SHGC and generalized cones whose axis is an arbitrary space curve. Rao and Medioni [61] derived the relationship between the GC descriptions and the surface descriptions. The gaussian and mean curvatures of GCs were derived and studied with respect to some useful geometrical properties. Rao and Medioni discussed the planarity and symmetry properties of the contour generators. They assumed that the imaging projection is orthographic. The types of GCs studied in their work are SHGC’s and GCs whose axis is an arbitrary space curve. Ponce and Chelberg [58] derived and proved several invariant properties of the contours of SHGC’s.

Two distinct and important types of contours that bound a surface, which are called extremal and discontinuity contours are proposed by Barrow and Tenenbaum [6]. An extremal contours are points on the curved surface where the viewed surface normal turns away smoothly from the viewer. Other terms having the same meaning which have been used by many researchers are contour generator [51] and limbs [57]. A discontinuity boundary occurs whenever the normal to the surface is discontinuous or at the intersection between surfaces. This is simply the two ends of a GC. Researchers have used the terms edges [58] and terminator [61] to describe the discontinuity boundary. In our work, contour generator and crease define respectively, the extremal and discontinuity contours.

Most of the existing methods used in analysing GCs assumed that the objects in the scene are being viewed under orthographic projection in which the lines of sight are parallel. In general vision systems, the camera may need to be moved closer to objects in the scene, in order to discriminate the subtle differences between objects or features. Under this viewing condition, perspective projection is more appropriate than orthographic. This is one of the main reasons which motivates us to derive some useful geometrical properties and equations of GC’s being viewed under perspective projection. These geometrical properties and features could be used for object recognition and shape restoration from an image. In the sequel, both quantitative and qualitative information about GC’s will be derived using the theory of differential geometry.

The main result of the work [73] reported in this chapter is the analytical expressions for the contour generators of SHGC’s and their projections under perspective, with the camera pointing at an
arbitrary point of interest. Two types of SHGC's, cylinder and frustrum are shown to yield as contour
generators a pair of straight lines. SHGC's with constant or linear sweeping function and an arbitrary
cross-section shape lead to a set of straight line contours under both orthographic and perspective
projection. It is also shown that from an arbitrary view point the projected contour generator of a
circular SHGC is not symmetric. However, under certain viewing conditions derived in the chapter
the projected contours become symmetric.

The practical significance of these results is multifold. First, occluding boundaries of objects
under perspective projection can be computed from the analytical contour generators at a fraction of
the time required for pointwise calculation. Second, the analytical form allows the user to predict
important object contour features such as points of high curvature. This is important for both a rapid
search for specified objects in the field of view and for dynamic (not precompiled) model invocation.
Third, it should be possible to use the contour generators to modify the point of interest of the
camera to restore contour symmetry and in this manner facilitate object recognition. This facility is
particularly relevant when vision is viewed as active process.

Our work is organised as follows: In Section 3.2, the basic definitions of both right and oblique
SHGC's are given. In Section 3.3, expressions for contour generators and their perspective projection
are derived and discussed. The mathematical framework for perspective projection of SHGC's with
respect to a general viewing coordinate system are discussed in detail. In Section 3.4, the properties
of circular SHGC are derived and summarized. In particular, plain cylinder (a SHGC with constant
sweeping rule) and cone (a SHGC with linear sweeping rule) are discussed in detail. In Section 3.5,
some useful properties of SHGC's with arbitrary cross-section and a constant or a linear sweeping
rule are discussed. In Section 3.6, the characteristics of our object modelling system are described in
moderate detail. The possible extensions of the modelling system with respect to model invocation
and geometric matching are proposed and described briefly.

3.2 A basic definition of an SHGC

An SHGC is a solid component which is generated by sweeping an arbitrary planar surface along
a straight axis. The sweeping rule can be constant, linear or non-linear. The eccentricity, the angle
between the straight axis and planar cross-sections of the SHGC is constant, but not necessarily
orthogonal. An orthonormal left-handed system (\( \hat{i}, \hat{j}, \hat{k} \)) is used as a local coordinate system for
defining points of an SHGC. The shape of a cross-section is defined by \( (u(\theta)\hat{i} + v(\theta)\hat{j}) \). The shape
and size of a GC are described by its sweeping function \( r(s) \). A point \( P(\theta,s) \) on the surface of an
SHGC can be represented by:

\[
P(\theta,s) = r(s) (u(\theta)\hat{i} + v(\theta)\hat{j}) + s\hat{k}
\]

\[
\theta \in [0, 2\pi]; \quad s \in [0, h]
\]
In more general case, an additional degree of freedom can be introduced into the orientation of the straight axis of an SHGC. Shafer [63] has introduced an angle of inclination between the cross-sections and axis of an SHGC. Ponce [58] has introduced three degrees of freedom to the axis of SHGC's. This class of GCs is called oblique SHGC under the taxonomy developed by Shafer [63]. A point on the surface of an oblique SHGC can be expressed as:

$$P(\theta, s) = (r(s)u(\theta) + s \cos \alpha_0 \sin \beta_0) \hat{i} + (r(s)v(\theta) + s \sin \alpha_0 \sin \beta_0) \hat{j} + s \cos \beta_0 \hat{k}$$

where \((\alpha_0, \beta_0)\) are the spherical coordinates of the slanted axis of an oblique SHGC. The representation of oblique SHGSs used in our work is similar to that described in Ponce's work [58].

3.3 Contour generators and their projection

3.3.1 Contour generators

Contour generators and their perspective projections are derived and studied in this section. Contour generators are points on the GC surface where the normal to the surface is orthogonal to the lines of sight. A general viewing condition is assumed in which perspective projection is used to analyse the properties of GCs. A perspective projection is characterized by a point known as the center of projection. In perspective projection, the intersections of lines from each point of a surface to a viewpoint with image plane determine the projected scene image. An example of perspective projection is depicted in Figure 3-1 (a).

We now derive an expression for the normal to the surface of an SHGC. The normal \(\hat{N}(\theta, s)\) to the surface of an SHGC at \(\hat{P}(\theta, s)\) is computed by the vector product of the two partial derivatives:

$$\frac{\partial \hat{P}(\theta, s)}{\partial \theta} = r(s)(\frac{\partial u(\theta)}{\partial \theta} \hat{i} + \frac{\partial v(\theta)}{\partial \theta} \hat{j})$$

$$\frac{\partial \hat{P}(\theta, s)}{\partial s} = \frac{\partial r(s)}{\partial s}(u(\theta) \hat{i} + v(\theta) \hat{j}) + \hat{k}$$

For the sake of brevity, we drop the argument \(\theta\) of \(u(\theta)\) and \(v(\theta)\), and \(s\) of \(r(s)\). Also \(\frac{\partial u(\theta)}{\partial \theta}\), \(\frac{\partial v(\theta)}{\partial \theta}\) and \(\frac{\partial r(s)}{\partial s}\) are denoted as \(u', v'\) and \(r'\) respectively. The non-normalised surface normal, \(\hat{N}(\theta, s)\) is expressed as:

$$\hat{N}(\theta, s) = \frac{\partial \hat{P}(\theta, s)}{\partial \theta} \times \frac{\partial \hat{P}(\theta, s)}{\partial s}$$

$$= (v' \hat{i} - u' \hat{j} + r'(vu' - uv') \hat{k})r$$

The spherical coordinates of a view point, \(V_v\) are \((\lambda, \alpha, \beta)\), where \(\lambda\) is a radial distance from the origin of the object coordinate frame \((\hat{i}, \hat{j}, \hat{k})\). \(V_v\) can be written:

$$V_v = e_1 \hat{i} + e_2 \hat{j} + e_3 \hat{k}$$
Analysis of SGHC

\[ v' \cos \alpha - u' \sin \alpha = \frac{w G_u}{\lambda \sin \beta} \]

where \( G_u = G_u(s) = C_n(1 - n)s^n + \sum_{i=0}^{n-1} [C_i(1 - i) + \lambda(i + 1)C_{i+1} \cos \beta] s^i \).

The contour generators for oblique SHGC can be derived using the same approach. By manipulating Eq (3.2) and Eq (3.3). The non-normalized surface normal of an oblique SHGC is expressed as:

\[ N(\theta, s) = [v' \cos \beta_0 i - u' \cos \beta_0 j + (u' \sin \beta_0 \sin \alpha_0 - v' \sin \beta_0 \cos \alpha_0 - r'w) k] r \]

The lines of sight from the center of projection, \( \vec{V}_e \), to each point on the surface of an oblique
SHGC are expressed as:

\[ V_{IS}(\theta, s) = (ur + s \sin \beta_0 \cos \alpha_0 - e_1) \hat{i} + (vr + s \sin \beta_0 \sin \alpha_0 - e_2) \hat{j} + (s \cos \beta_0 - e_3) \hat{k} \]

The contour generators of the oblique SHGC are defined as:

\[ w[r' (\lambda \cos \beta - s \cos \beta_0) + r \cos \beta_0] + \lambda (u' g_1 - v' g_2) = 0 \quad (3.5) \]

where

\[ g_1 = \sin \alpha \sin \beta \cos \beta_0 - \cos \beta \sin \beta_0 \sin \alpha \quad \text{and} \quad g_2 = \cos \alpha \sin \beta \cos \beta_0 - \sin \beta_0 \cos \alpha \cos \beta \]

For a sweeping function expressed as a polynomial of degree \( n \), Eq (3.5) can be rewritten as:

\[ v' g_2 - u' g_1 = \frac{w G_0}{\lambda} \quad (3.6) \]

where \( G_0 = G_0(s) = (1 - n)C_n \cos \beta_0 s^n + \sum_{i=0}^{i=n-1} [(1 - i)C_i \cos \beta_0 + \lambda(i + 1)C_{i+1} \cos \beta] s^i \).

### 3.3.2 Projection of contour generators

An expression for the perspective projection of contour generators will be derived and studied in the following section. First, we introduce a general and realistic viewing coordinate system, in which the camera or viewpoint, \( V_e = (e_1, e_2, e_3) \), can be pointed in any viewing direction specified by a point of interest, \( V_p = (p_1, p_2, p_3) \) with respect to the object coordinate frame. An image plane is normal to the \( z \)-axis which is aiming at the point of interest. The focal length, \( f \) is the normal distance between an image plane and a viewpoint. The viewing coordinate system is defined by the viewing direction \( \hat{v} \) and the vector basis of the image plane (\( \hat{w}, \hat{u} \)). The arrangement of the viewing coordinate system is shown in Figure 3-1 (b). The 3D coordinates \((X_v, Y_v, Z_v)\) of a point \((X_w, Y_w, Z_w)\) with respect to the viewing coordinate system can be related by a transformation matrix \( T_{\text{view}} \) as \[69\]

\[
\begin{pmatrix}
X_v \\
Y_v \\
Z_v \\
1
\end{pmatrix} = [T_{\text{view}}] \begin{pmatrix}
X_w \\
Y_w \\
Z_w \\
1
\end{pmatrix}
\quad (3.7)
\]

First, the origin of the world coordinate system is translated to \( V_e \) using a translation transformation
Analysis of SGHC

defined as :

\[
[T_i] = \begin{pmatrix}
1 & 0 & 0 & -e_1 \\
0 & 1 & 0 & -e_2 \\
0 & 0 & 1 & -e_3 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

Second, the translated coordinate system is orientated in such a way that the vector \( \vec{V} \) (Z-axis) of the viewing coordinate system is aimed at the point of interest. The rotation transformation is defined as :

\[
[T_{Ro}] = \begin{pmatrix}
w_1 & w_2 & w_3 & 0 \\
u_1 & u_2 & u_3 & 0 \\
v_1 & v_2 & v_3 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

where, \( \vec{v} = v_1 \hat{i} + v_2 \hat{j} + v_3 \hat{k} \), is a viewing direction, and \( \vec{w} = w_1 \hat{i} + w_2 \hat{j} + w_3 \hat{k} \) and \( \vec{u} = u_1 \hat{i} + u_2 \hat{j} + u_3 \hat{k} \) are the X- and Y- axis of the image plane respectively. The orthogonal vectors of the viewing coordinate frame are defined as :

\[
\vec{v} = \frac{p_1 - e_1}{V_m} \hat{i} + \frac{p_2 - e_2}{V_m} \hat{j} + \frac{p_3 - e_3}{V_m} \hat{k}
\]

\[
\vec{w} = \frac{p_2 e_3 - p_3 e_2}{W_m} \hat{i} + \frac{p_3 e_1 - p_1 e_3}{W_m} \hat{j} + \frac{p_1 e_2 - p_2 e_1}{W_m} \hat{k}
\]

\[
\vec{u} = \frac{\vec{w} \times \vec{v}}{||\vec{w} \times \vec{v}||}
\]

where

\[
V_m = \sqrt{(p_1 - e_1)^2 + (p_2 - e_2)^2 + (p_3 - e_3)^2}
\]

\[
W_m = \sqrt{(p_2 e_3 - p_3 e_2)^2 + (p_3 e_1 - p_1 e_3)^2 + (p_1 e_2 - p_2 e_1)^2}
\]

Finally, the camera is rotated about \( \vec{v} \) (Z-axis) and the direction of the Y-axis in the viewplane. The transformation for rotating the coordinate frame about the viewing direction \( \vec{v} \) in clockwise direction is defined as :

\[
[T_{Rv}] = \begin{pmatrix}
cos \phi & sin \phi & 0 & 0 \\
-sin \phi & cos \phi & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]
It can be easily shown that in the case of Figure 3-1 (b), \( T_{\text{view}} \) is defined as \( T_{\text{view}} = T_RuT_{Ro}T_I \):

\[
[T_{\text{view}}] = \begin{pmatrix}
w_1 \cos \phi + u_1 \sin \phi & w_2 \cos \phi + u_2 \sin \phi & w_3 \cos \phi + u_3 \sin \phi & -k_1 \cos \phi - k_2 \sin \phi \\
u_1 \cos \phi - w_1 \sin \phi & u_2 \cos \phi - w_2 \sin \phi & u_3 \cos \phi - w_3 \sin \phi & k_1 \sin \phi - k_2 \cos \phi \\
v_1 & v_2 & v_3 & -k_3 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

\( k_1 = V_e \cdot \hat{v}, \quad k_2 = V_e \cdot \hat{u} \) and \( k_3 = V_e \cdot \hat{v} \) are the dot products of a position vector and orthogonal vector basis of the viewing coordinate system. \( \phi \) is an angle of rotation about the viewing direction \( \hat{v} \) in clockwise direction.

The projections of SHGC contour generators can then be written:

\[
X_I = \frac{X_v}{Z_v}; \quad Y_I = \frac{Y_v}{Z_v};
\]

### 3.4 Analysis of circular SHGC's or solids of revolution

In this section, we will derive and study the contour generators and the projected contours of circular SHGC's (CSHGC). For a CSHGC, \( u = \cos \theta, v = \sin \theta \) and \( w = 1 \), and the contour generators can be written:

\[
P(\theta, s) = r(\cos \theta \hat{t} + \sin \theta \hat{j}) + s \hat{k}
\]  

From Eq (3.4), the contour generators of a CSHGC can be expressed as:

\[
\cos(\theta - \alpha) = F_u
\]  

where \( F_u = \frac{G_u}{r \sin \theta} \). The solutions of the contour generators are \( \theta = \alpha \pm \cos^{-1}(F_u) \). By substituting these values into Eq (3.7) and Eq (3.8), the projection of the contour generators can be expressed as:

\[
\begin{pmatrix}
X_v \\
Y_v \\
Z_v \\
1
\end{pmatrix} = [T_{\text{view}}] \begin{pmatrix}
r F_u \cos \alpha \pm r \sin \alpha \sqrt{1 - F_u^2} \\
r F_u \sin \alpha \pm r \cos \alpha \sqrt{1 - F_u^2} \\
s \\
1
\end{pmatrix}
\]

\[
X_I = f \frac{X_v}{Z_v}; \quad Y_I = f \frac{Y_v}{Z_v};
\]

From Eq (3.6) and Eq (3.7), the solutions of the contour generators for oblique CSHGC can be
expressed as:

$$\cos(\theta - \gamma) = F_0$$  \hfill (3.11)

where $F_0 = \frac{G_1}{\lambda \sqrt{G_{11} + G_{22}}}$ and $\gamma = \tan^{-1}\left(\frac{G_1}{G_2}\right)$. The projection of the oblique CSHGC can be expressed:

$$
\begin{pmatrix}
X_v \\
Y_v \\
Z_v \\
1
\end{pmatrix}
= [T_{\text{view}}]
\begin{pmatrix}
r F_0 \cos \gamma + r \sin \gamma \sqrt{1 - F_0^2 + s \sin \beta_0 \cos \alpha_0} \\
r F_0 \sin \gamma \pm r \cos \gamma \sqrt{1 - F_0^2 + s \sin \beta_0 \sin \alpha_0} \\
s \cos \beta_0 \\
1
\end{pmatrix}
$$

$$
X_I = f \frac{X_v}{Z_v}; \quad Y_I = f \frac{Y_v}{Z_v}
\hfill (3.12)
$$

3.4.1 Symmetry of Projections

Symmetry property of projections of contour generators may have an important role in shape restoration and object recognition. A contour is said to be symmetric iff the point-wise correspondences of the contour are at equal distance from the projected axis, and the line segments joining the point-wise correspondences are orthogonal to the projected axis of the CSHGC. A projected axis is known as the axis of symmetry if the contours are symmetrical about the projected axis. In perspective projection, the contours of a CSHGC are generally not symmetric but under certain viewing conditions defined in the following lemma they will be:

**Lemma 1**: In perspective projection, the projected contours of a circular SHGC with an arbitrary sweeping function are symmetrical about its projected axis iff the image plane of the viewing coordinate system is perpendicular to the plane passing through the axis of the CSHGC and a viewpoint (see Figure 3-2).

This viewing condition can be achieved by choosing the point of interest $V_p = (p_1, p_2, p_3)$ as the point of intersection on the plane passing through the axis of the CSHGC and the viewpoint $V_e$. There is no loss of generality in assigning $\phi$ to 0. The projection of a 3D point on a 2D image, is defined as:

$$
X_I(\theta) = \frac{f}{Z(\theta)} \left[ r(w_1 \cos \theta + w_2 \sin \theta) + w_3 s \right]
$$

$$
Y_I(\theta) = \frac{f}{Z(\theta)} \left[ r(e_2(e_2 - e_3) - p_3) \cos \theta + e_1(p_3 - p_1 e_3) \sin \theta) + s(p_1 e_2 - p_2 e_1) \right]
$$

Under the viewing condition $(p_1 e_2 = p_2 e_1)$ described in *Lemma 1*, the projection of the contour can...
be simplified as:

\[ X_t(\theta) = \frac{fr(e_1 p_3 - p_1 e_3)}{W_m Z(\theta) \cos \alpha} \sin(\theta - \alpha) \]  
\[ Y_t(\theta) = \left[ \frac{r(p_3 - e_3)(e_1 p_3 - p_1 e_3)}{W_m V_m \cos \alpha} \right] \frac{f}{Z(\theta)} \cos(\theta - \alpha) + sv_3 - k_3 \]  

where,

\[ Z(\theta) = \left[ \frac{r(p_1 - e_1) \cos(\theta - \alpha) + sv_3 - k_3}{V_m \cos \alpha} \right] \]
\[ V_m = \sqrt{(1 - \frac{e_1}{p_1})^2 (p_1^2 + p_2^2) + (p_3 - e_3)^2} \]
\[ W_m = \sqrt{(p_2 e_3 - p_3 e_2)^2 + (p_3 e_1 - p_1 e_3)^2} \]

By substituting the expressions of \( \theta \) derived from Eq (3.9) into Eq (3.13) and Eq (3.14), the image coordinates of the contours can be written:

\[ P_t(s) = \pm X_c(s) \hat{w} + Y_c(s) \hat{u} \]

The vector of the line segment joining the point-wise contours is \( 2X_c(s) \hat{w} \), which is perpendicular to the \( u \)-axis of the image plane (i.e. \( 2X_c(s) \hat{w} \cdot \hat{u} = 0 \)). The distance from the point-wise correspondences to the projected axis (\( u \)-axis in this case) is equal to \( X_c(s) \). This proves the lemma.

**3.4.2 CSHGC's with Constant or Linear Sweeping Function**

For a given CSHGC with constant sweeping function (i.e. \( r = \) constant). The degree of the polynomial sweeping function is zero (\( n = 0 \)). By substituting \( n = 0 \) into Eq (3.9), the solutions of the contour generators can be written:

\[ \theta = \alpha \pm \cos^{-1} \left( \frac{C_0}{\lambda \sin \beta} \right) \]

The contour generators are a pair of 3D parallel lines. Similarly, from Eq (3.11) the contour generators of an oblique CSHGC with constant sweeping function can be expressed as:

\[ \theta = \tan^{-1} \left( \frac{\bar{g}_1}{\bar{g}_2} \right) \pm \cos^{-1} \left( \frac{C_0 \cos \beta_0}{\lambda \sqrt{\bar{g}_1^2 + \bar{g}_2^2}} \right) \]

Again, \( \theta \) is independent of \( s \). The contour generators of an oblique CSHGC are a pair of 3D parallel lines. Thus, the following lemma can be stated:
Lemma 2: In perspective projection, the contour generators of a right or oblique CSHGC with constant sweeping rule are a pair of parallel straight lines. The perspective projection of the right CSHGC contour generators are a pair of projected 2D parallel straight lines iff the image plane is parallel to the axis of the CSHGC. Otherwise, the contours appear as a pair of convergent lines. In general, the perspective projection of the oblique CSHGC contour generators is a pair of convergent lines (see Figure 3-3).

For a given CSHGC with linear function \( r = C_0 + C_1 s \), the GC is usually known as a cone or frustrum. When the degree of the sweeping function is one \((n=1)\) the solution of contour generators in Eq (3.9) becomes:

\[
\theta = \alpha \pm \cos^{-1} \left( \frac{C_0 + \lambda C_1 \cos \beta}{\lambda \sin \beta} \right)
\]

The contours are a pair of straight lines which are parallel to the axis of the linear CSHGC. Similarly, from Eq (3.11), the solution of contour generators of an oblique CSHGC with linear sweeping function can be expressed as:

\[
\theta = \tan^{-1} \left( \frac{\theta_1}{\theta_2} \right) \pm \cos^{-1} \left( \frac{C_0 \cos \beta_0 + \lambda C_1 \cos \beta}{\lambda \sqrt{\theta_1^2 + \theta_2^2}} \right)
\]

The results can be summarised in the following lemma:

Lemma 3: For a given right or oblique CSHGC with a linear sweeping function, the contour generators are a pair of 3D convergent lines. If the convergent line contour generators of a right CSHGC are extended they will intersect the axis of the GC. This is not true for the case of oblique CSHGC (see Figure 3-4).

3.5 Analysis of SHGC's with constant or linear sweeping function

The cross-section of a SHGC is described by \( r(u \ i + v \ j) \). For clarity, \( u \) and \( v \) are replaced by \( \rho \cos \theta \) and \( \rho \sin \theta \) respectively, where \( \rho = \rho(\theta) \) is a parametric function describing the reference cross-section. From Eq (3.4), the contour generators can be described by:

\[
\cos(\theta - \psi - \alpha) = \frac{\rho'^2 G_u}{\lambda \sin \beta \sqrt{\rho'^2 + \rho^2}}
\]

where \( \rho' = \frac{d\rho(\theta)}{d\theta} \) and \( \psi = \tan^{-1} \frac{\rho'}{\rho} \). For a SHGC with an arbitrary cross-section described by \( \rho(\cos \theta \ i + \sin \theta \ j) \), with a constant sweeping rule, the value of \( G_u \) is constant, (i.e. \( G_u = C_0 \)). The solution of Eq (3.15) describing the contour generators with a constant sweeping rule are some constants \( \theta = \theta_u \), independent of \( s \). The parametric function, \( P(\theta_u, s) \), of the contour generator of
the corresponding SHGC can be written:

\[ P(\theta_u, s) = C_0 \rho(\theta_u) \left[ \cos(\theta_u) \hat{l} + \sin(\theta_u) \hat{j} \right] + s \hat{k} \]  

(3.16)

From the equation, it is apparent that the contour generator of a SHGC, with an arbitrary cross-section and a constant sweeping rule is a list of 3D parallel straight lines.

Similarly, for a right SHGC's with arbitrary cross-section and linear sweeping rule \( G_u = C_0 + \lambda C_1 \cos \beta \), the solutions of Eq (3.15) are some constants \( \theta = \theta_u^l \), independent of \( s \). The contour generators can be written:

\[ P(\theta_u, s) = (C_0 + C_1 s) \rho(\theta_u) \left[ \cos(\theta_u) \hat{l} + \sin(\theta_u) \hat{j} \right] + s \hat{k} \]  

(3.17)

The contour generators are a list of convergent straight lines. The tangents of the contour generators can be expressed as:

\[ \hat{t}(\theta_u) = C_1 \rho(\theta_u) \left[ \cos(\theta_u) \hat{l} + \sin(\theta_u) \hat{j} \right] + \hat{k} \]  

(3.18)

A point along the tangent line can be written as:

\[ \bar{P}(\theta_u, s) + m\hat{t} = (C_1 s + C_0 + mC_1) \rho(\theta_u) \left[ \cos(\theta_u) \hat{l} + \sin(\theta_u) \hat{j} \right] + (s + m) \hat{k} \]  

(3.19)

where \( m \) is a constant. The point of intersection between the tangent line and axis of the SHGC is equal to \(-\frac{C_0}{C_1} \hat{k}\), which is independent of \( s \) and \( \theta \). The result shows that the tangents of the 3D convergent lines intersect at the same point on the axis. These results can be summarized in the following lemma.

**Lemma 4:** In perspective projection, the contour generator of a right SHGC with an arbitrary and a constant or a linear sweeping rule is a list of straight lines. In the case of constant sweeping function, the contour generator is 3D parallel straight lines. The projected 2D straight lines are parallel if the image plane is parallel to the axis of the G.C. For a linear sweeping function, the contour generator is a list of convergent straight lines of which the tangent lines intersect at the same point on the axis of the SHGC (see Figure 3-5).

For an oblique SHGC with an arbitrary cross-section and a constant sweeping rule, the solutions of the contour generators, from Eq (3.6), can be written:

\[ \cos(\theta - \psi - \delta) = \frac{\rho^2 G_o}{\lambda \sqrt{(\rho^2 + \rho^2)(g_1^2 + g_2^2)}} \]  

(3.20)

where \( \delta = \tan^{-1} \frac{g_1}{g_2} \). From Eq (3.20), the solutions of the contour generators for an oblique SHGC with constant sweeping rule, \( G_o = C_0 \cos \beta_0 \), are constants \( \theta = \theta_o^c \) independent of \( s \). Thus, the
Analysis of SGHC

Contour generator of an oblique SHGC can be written:

\[
P(\theta_o, s) = (C_0 \rho(\theta_o) \cos(\theta_o) + \cos \alpha_0 \sin \beta_0 s) \hat{i} \\
+ (C_0 \rho(\theta_o) \sin(\theta_o) + \sin \alpha_0 \sin \beta_0 s) \hat{j} + s \cos \beta_0 \hat{k}
\] (3.21)

The tangents of the contour generator are:

\[
\hat{t}(\theta_o) = \cos \alpha_0 \sin \beta_0 \hat{i} + \sin \alpha_0 \sin \beta_0 \hat{j} + \cos \beta_0 \hat{k}
\] (3.22)

The tangents of the contour generator are parallel to the slanted axis of the oblique SHGC.

Similarly, for an oblique SHGC with a linear sweeping rule, \( G_o = C_0 \cos \beta_0 + \lambda C_1 \cos \beta \), the contour generator can be written as:

\[
P(\theta_o, s) = [C_0 \rho(\theta_o) \cos(\theta_o) + (C_1 \rho(\theta_o) \cos(\theta_o) + \cos \alpha_0 \sin \beta_0 s) \hat{i} \\
+ [C_0 \rho(\theta_o) \sin(\theta_o) + (C_1 \rho(\theta_o) \sin(\theta_o) + \sin \alpha_0 \sin \beta_0 s) \hat{j} + s \cos \beta_0 \hat{k}
\] (3.23)

The tangents of the contour generators can be written:

\[
\hat{t}(\theta_o) = (C_1 \rho(\theta_o) \cos(\theta_o) + \cos \alpha_0 \sin \beta_0) \hat{i} \\
+ (C_1 \rho(\theta_o) \sin(\theta_o) + \sin \alpha_0 \sin \beta_0) \hat{j} + \cos \beta_0 \hat{k}
\] (3.24)

The contour generator is a list of straight lines. These results can be summarized in the following lemma:

**Lemma 5**: In perspective projection, the contour generator of an oblique SHGC with a constant or a linear sweeping function is a list of straight lines. For a constant sweeping rule, the contour generator is a list of straight lines which are parallel to the slanted axis of the oblique SHGC. In general, the projected 2D straight line contours of the oblique SHGC are not parallel. For a linear sweeping rule, the 3D straight lines do not intersect at the same point on the axis of the SHGC (see Figure 3-6).

### 3.6 Application of GCs

Currently, a generic vision system is being investigated and developed for scene interpretation. This vision system will contain models of both polyhedral and curved objects. A preliminary version of an object modelling system using G.C. as representational primitives has been implemented. Both 2D and 3D geometric features such as contour generators, projected 2D contours, crease and curvatures can be derived from the modelling system. In the modelling system, GCs. represent the primitive components or parts of which more complex objects are built. Each G.C. is specified with respect to its own coordinate system and then each G.C. can be transformed in order to abut several together to form an object specified in a global coordinate system. Six modelling examples are illustrated in
Analysis of SGHC

Figure 3-7. The objects depicted in Figure 3-7(a) to 3-7(d) are represented by the composition of SHGC primitives. Their projections are analytically computed using the approach discussed herein. The handle of the milk jug (see Figure3-7(e)) is represented by a G.C with a 3D space curve axis. The projected contours of the milk jug handle are extracted using pointwise calculation. A prototype of non-straight axis G.C. primitive and its projection are shown in Figure 3-7(f).

Three main issues will be closely considered in future work: the organisation of model descriptions in a hierarchical structure; the development of strategies and techniques for model invocation; the implementation of symbolic and geometric model-based matching using both 2D and 3D features derived from the solid modelling system.

The object model descriptions will be organised into symbolic or qualitative and geometric or quantitative level. In symbolic level, viewpoint-invariant, salient and distinctive features are stored and used for model invocation and reduction of the search space in matching. In quantitative description level, geometric features such as contour generators, 2D projected contours, curvatures and junction types formed between contours and crease will be evaluated and used for both 2D and 3D geometric matching.

The vision system has to cope with models of many complex objects. Consequently, the matching of each model against all the scene data in order to get the best scene interpretation is generally infeasible. Instead, methods are being developed to reliably group data and image descriptions into meaningful structures such as parallelism, colinearity, lines in proximity, circular arcs and high-curvature points, etc so that only a small number of candidates are investigated in detail.

There are many potential sources of knowledge which can be exploited for both model invocation and geometric matching. Some examples are G.C. primitive relations such as the angles between the axes of GCs. and the location where GCs are abutted can be used to constrain the prediction of the orientation of objects in the scene. High-level symbolic features such as the contour generators of SHGC's which are either convergent straight lines, parallel straight lines, lines of symmetry, circular or elliptical creases etc. can be used as cues for model invocation.

For example, Figure(3-8) shows that G.C.(A), (B) and (C) are selected for more detailed examination if straight lines of plausible length are detected from the scene descriptions. G.C. (C) can be removed from any further consideration if the extracted 2D straight lines are parallel. To select a single model from the two candidates and estimate its parameter such as G.C. pose, we can use the junction types formed by the straight lines and the creases or the descriptions of the 2D projected creases etc. In this case, G.C. (B) will be chosen for quantitative matching if a 2D projected elliptical shape is extracted from the scene image.

A manageable size of object model set will be invoked from the model-base using observed features and their combinations. To perform a detailed check of the correspondences between inferred models and image data, different geometric matching strategies using both 2D and 3D information will be invoked for each type of feature extracted from the image. The qualities and types of features computed from a grey-intensity image will be dependent on the sophistication and effectiveness of...
the low-level imaging processes. The types of geometric features that can be extracted from an
SHGC for matching are 2D projected contours [20], the relations between projected contours and
creases [36], junctions [34], curves [45] and curvatures [53]. A decomposition-approach for matching
will be developed and integrated using the existing techniques. This approach will be preceded by
matching decomposed subparts of objects represented by GCs and the aggregated result of these
subpart matches will determine the confidence of an object interpretation.

3.7 Conclusions

In this chapter a mathematical framework for analysing SHGC's under perspective has been devel-
oped. Using this framework a number of results concerning geometric properties of SHGC's have
been derived. In particular, the right and oblique SHGC are analysed in detail. The expressions for
contour generators of SHGC's and their projection have been derived and studied. Two types of
SHGC, cylinder and frustrum (subset of cone) have been found to yield as contour generators, a pair
of straight lines. SHGC's with constant or linear sweeping function and of arbitrary cross-section
shape lead to a list of straight line contours under both orthographic and perspective projections. An
elliptical cross-section SHGC which is a simple example of this class of GC has been studied in
detail. From an arbitrary view point, the projected contour generator of an SHGC is not symmetric
about its projected axis. Our future work will concentrate on the derivation of other useful 2D and
3D geometrical features such as curvature and surface patches for a larger class of GCs in perspective
projection. Both 2D and 3D salient, distinctive and viewpoint-invariant geometric features will be
determined from GCs. These features will be used for both 2D and 3D geometrical matching. The
properties of GCs will be exploited in modelling objects composed of GC primitives.
Figure 3-1: (a) A perspective projection of an SHGC. (b) A viewing coordinate system.

Figure 3-2: The projected contours of a CSHGC with an arbitrary sweeping function is symmetrical about its projected axis.
Figure 3-3: The contour generators of a right or oblique CSHGC with a constant sweeping rule are a pair of parallel straight lines.

Figure 3-4: The contour generators of a right or oblique CSHGC with a linear sweeping function are a pair of 3D convergent lines.
Analysis of SGHC

Figure 3-5: The contour generator of a right SHGC with an arbitrary and a constant or a linear sweeping rule is a list of straight lines.

Figure 3-6: The contour generator of an oblique SHGC with a constant or a linear sweeping rule is a list of straight lines.
Analysis of SGHC

Figure 3-7: Examples of object models

(a) A glass
(b) A wineglass
(c) A spoon
(d) A fork
(e) A milk jug
(f) A G.C. with an arbitrary 3D curve axis

Figure 3-8: A simple example of decomposed subparts model invocation
Chapter 4

Feature Extraction

4.1 Introduction

Model-based recognition of three-dimensional objects relies largely on explicit geometric object models and sensed data to construct a scene description. Vision system based on analysis of two-dimensional line segments extracted from a single gray-intensity image is a difficult task because the explicit depth information is not preserved during the process of projection. Moreover, the line drawing produced in the pre-processing is generally incomplete and noisy as a result of occlusion, shadows, surface texture and complicated lighting. An issue which is of particular concern with matching 3D models to 2D image data is the variation of object feature appearance as a function of viewing position. In the stage of hypothesis generation, the pose of scene objects is unknown and exhaustive matching to all possible 2D projections is computationally too expensive. However there are important 2D cues which are indicative of specific 3D structures. In particular, an empty area bounded by three or more linear segments, i.e. an open or closed polygon, is most simply interpreted as a flat surface. Thus the recognition of polygonal image curves provides important information which can be used in model-based recognition.

The ACRONYM (Brooks [16]) vision system clearly demonstrated the potential power of geometric models to constrain 3D object recognition and was influential in coining the term model-base vision. Around the same time other work was being pursued on the exploitation of geometry. Horaud and Bolles [33] and, Bolles and Cain [13] developed the local focus feature idea whereby a group features could be used as a cue into a 2D model. Lowe [50] experimented with groups of features in his SCERPO system and utilised the idea of perceptual grouping i.e non-accidental structures which possess a degree of viewpoint invariance. Many systems were built around variants of these ideas using other types of compound primitives.

The limitations of the current existing system methods are: the efficiency of the system decrease substantially when the knowledge-base of the object models is very large; the systems are very sensitive to object occlusion. The goal of our preliminary vision system is to solve these problems within the domain of polyhedral objects. Our work is motivated by Lowe's model-based vision
Feature Extraction

system [50] using the paradigm of perceptual organisation to bridge the gap between 2D images and knowledge of 3D objects without any depth information. In our vision system, a method is developed to generate higher-order features such as opened and closed polygonal chains and their relationships using intermediate features such as collinear lines and junctions. The qualitative geometric cues of the higher-order features are exploited as sources of evidence for hypothesizing plausible objects.

Having hypothesized a few number of plausible object models using coarse and abstract information, the position and orientation of a camera with respect to a scene object is determined by establishing the correspondences between image and object models. This combinatorics of possible assignments of low-level or intermediate primitive features to model features can be explosive and practically infeasible. The complexity of the searching space can be reduced significantly using the connectivity and geometric constraints between related higher-order features extracted from the process of perceptual grouping.

In this chapter, the perceptual grouping of higher-order features using intermediate features such as collinear lines and junctions is emphasized and discussed in details. We present a new algorithm for grouping 2D line segments into open and closed polygons that correspond to feasible physical 3D structures (Wong [74]). The algorithm starts by identifying junctions made of two line segments and then forms triples by combining pairs of junctions which share a common line. These triples are then scanned by a procedure which connects them into polygon structures. Heuristic rules are used to control the combinatorial explosion associated with unconstrained associations of junctions and triples. Physical rules are used to reject polygons which are incompatible with a single planar surface hypothesis. The algorithm does not require strict connectivity of end-points at junctions. The polygon finder is seen as a high level grouping step which will enable the recognition of 3D objects which contain flat polygonal faces. Its use is illustrated on real data. The practical significant of these higher-order features is discussed. The potential sources of knowledge which can be exploited in hypothesis generation and geometric matching of our experimental vision system are demonstrated in the next two chapters.

4.2 Junctions and triples

The polygon finding algorithm begins with a set of straight line segments extracted from an image. This can be achieved from gray level data by using edge finding followed by a Hough-based line finder (Princen [60]). A basic concept in the algorithm is that of a junction region. The definition adopted follows the work of Horaud and Veillon [38] on perceptual grouping. A junction region is an image window which contains one or more line segment end-points. The maximum size of a junction region is predefined and line segment comparisons are made to determine whether line segments endpoints are mutually proximal to within this value. Simple geometric tests can be used to limit the comparisons which are necessary to determine this. Each junction region is given a unique integer identifier, $R_i$, and line segments are then designated as $l_{ij}$ where the subscripts relate to the
Feature Extraction

Figure 4-1: An example of junction configurations

junction region labels of their constituent endpoints, see Figure 4-1.

The next step of the algorithm proceeds by considering merging line-pairs which share a common junction point and thereby constitute a V. Pairs of V-junctions are grouped to form “triple” line structures subject to certain constraints. The use of triples was recently suggested by Henikoff and Shapiro [31] but in our work we refine the concept by introducing a distinction between “weak” and “plausible” V-junction types. A “weak” V-junction is one in which the interior angle subtended by the two lines includes a third line segment. In Fig. 4-1 the V-junction formed by the pair \{l_{02}, l_{23}\} is a weak junction while \{l_{23}, l_{43}\} is a plausible junction. The importance of the distinction is that the plausible junctions are more likely to be representative of significant physical structure and therefore although all V junction types are considered, the plausible junctions are considered first in the search process. This ordering reduces the computational expense of the search process.

In forming triples the two merging V-junctions must have a common line segment and their resultant triple must have only two endpoints. The triple is called a “weak” triple if either of the constituent junctions is a “weak” junction. A “plausible” triple is formed using two “plausible” V-junctions. In many cases there are several V-junctions at each junction region and it is then necessary to choose to combine into a triple only that V-junction whose subsequent continuation is likely to yield a physically significant closed polygon. Arbitrary investigation of all possible continuations is computationally expensive and therefore heuristics are used to limit this process. To appreciate these heuristics consider Figure 4-2(a) which shows a situation where there are two possible V-junctions \{l_{k2,k1}, l_{2,k1}\} and \{l_{k2,k1}, l_{k2,k2}\} that might be merged with junction \{l_{k1,k}, l_{k1,k2}\} to form triples. The common line of the triple, \!l_{k2,k1}! is called the linking segment and its extension divides the 2D plane into two half-planes. Line \!l_{k1,k1}! is known as the seed segment while the remaining lines are possible branching segments. The choice of V-junction to include in the triple can be decided by considering the angular relationship between the seed segment and the branching segment and whether their endpoints lie in the same or different half planes. Only three distinct situations need be considered. In a case where the two branching endpoints lie in different half-planes then the best branch to choose
Figure 4-2: (a) Two possible interpretations of triples (b) Only one of the two possible branching points is located on the same side of a linking segment (c) Two possible branching points are located on the same side of a linking segment (d) Two possible branching points are located on the different side of a linking segment.

is that one whose endpoint is in the same half-plane as the seed segment endpoint, see Figure 4-2(b). If both branching segment endpoints lie in the same half-plane as the seed segment endpoint then the segment to choose is that one whose dot-product with the linking segment is largest, see Figure 4-2(c). By contrast, if both branching segment endpoints lie in the half plane opposite the seed segment endpoint then it is desirable to choose the branching segment whose dot-product with the linking segment is smallest, see Figure 4-2(d).

A further heuristic which is important in constraining the search effort is the identification of "isolated triples". These are triples whose two interior junction regions consist of only the two segments which take part in the triple. Isolated triples are extremely significant and therefore in the merging processes which form triples into polygons they are considered early in the search.

4.3 Merging to form 2D Polygons.

The result of the processing discussed in the last section is a list of triples which are likely to be part of a closed polygon structure. The next part of the algorithm is concerned with the linking of these triples into higher order structures to form these polygons. Once again specific heuristics are used to guide the search to a successful conclusion. Firstly, those single triples whose two endpoints are in the same junction region are identified as triangles and removed from further consideration. Secondly, quadrilaterals are identified as formed from two overlapping triples which share a common endpoint junction regions. These too are removed from consideration before the search is initiated for five or more sided polygons.

In looking for significant polygon chains an arbitrary starting triple is chosen. The algorithm attempts to extend this triple by following a chain of triples around in a loop in a single direction.
The initial search direction is called the forward chaining direction. If attempts to extend the chain in the forward direction fail then the search attempts to extend the chain by starting at the other end and going in the opposite direction. This is known as the reverse chaining direction. To help explain the process in detail we introduce some terminology, which is illustrated in Figure 4-3. The last segment which is part of the chain formed by forward chaining is known as the forward chain header. The second last segment of the forward chain is termed the forward chain pre-header. The isolated endpoint of the forward chain header segment is called the forward leading point. Similar terms apply to the segments and points associated with the reverse chain direction.

When adding a triple to a partial chain three possible situations can be distinguished depending on the overlap between the chain and the triple. These possibilities are known as single point (sp), single segment (ss) and double segment (ds) merges. These are illustrated in Figure 4-4. In a single point merge the endpoint of the merging triple coincides with the leading point of the chain. In single segment merging the seed or branch segment of the triple and the chain header are the same segment. Finally, a double segment merge occurs when two lines of the triple are shared with the header and pre-header of the partial chain. At each point of the merging operation there may be for a given partial chain and each merge type several choices for the triple to be appended. For each choice, a closed chain test is first carried out. For example, in Figure 4-3 when the double segment merging triple is added to the chain the end point $P_{i+2}$ is compared with all points preceding $P_{i+2}$. If the end point is within the size of a junction region from one of these points then a closed chain is identified. A similar test applies in the case of single segment merging. If there is only one choice then that triple is known as a unique merging triple. Unique merging triples are particularly relevant in the formation of significant polygon chains and therefore a heuristic which is used to effectively limit the search process is that we initially attempt to extend the chain using only unique merging triples.
Figure 4-4: (a) An unique single point merging triple (b) An unique single segment merging triple (c) An unique double segment merging triple.

Unique single and double segment merging triples are searched for in the forward direction and if this fails then the search moves in the reverse direction. If a unique merging triple cannot be found in either direction then a further strategy of looking for a bridging chain is employed. The bridging operation involves extending the chain at both ends by considering all single point merges and all single segment merges. Double segment merges are not relevant for this bridging process. A closed chain is then identified if the extended chains overlap one another. At any stage of the merging process one of three outcomes may occur:

1. a closed chain may be identified.
2. a chain whose endpoints are isolated may be identified. This is called an open chain.
3. a chain may have potential triples for its extension but none conform with the unique merging or bridging rules discussed above. In this case the chain is said to be unresolved.

In a first pass through the triple data the set of triples which have to be considered in the process is pruned down by removing those which have two consecutive segments shared with an identified plausible closed chain. Triples which are found to be part of open or unresolved polygon chains are not removed from consideration at this stage as they may still be part of other, yet to be discovered, closed chains. A second pass through the reduced data is then done to identify unambiguously the open chains. A feasible open chain should have only one segment shared with a closed chain, hence it is best to use triples which share a middle segment with a segment of a closed chain as a starting triple for initiating the chaining process. An open chain extracted by the grouping process is described by its number of segments and the type of terminating condition. The end point of an open chain can be described by an element of the set $P = \{e, t, u\}$, where $e$ denotes an isolated end point, $t$ denotes the tail of a lambda junction and $u$ denotes an uncertain point where several lines may emanate from the junction. Some examples are illustrated in Figure 4-5. In Figure 4-5(b), the darkened chain is described as a feasible open chain having 4 segments and its ends are terminated at an uncertain
point and at the tail of a lamda junction. Throughout the triple formation and merging processes the rules of combination have been formulated so that the list of candidate triples is swiftly pruned and hence the computational cost is kept low.

4.4 Selection of physically feasible polygons

As stated in the introduction the detection of polygons is motivated by the desire to interpret the enclosed regions as single planar surfaces. The majority of 2D polygons detected by the algorithm discussed in the last section represent such physically feasible hypotheses. However some cases correspond to situations which are not physically feasible as single planes. Fortunately these can be rejected by incorporating further more specific, physically based rules into the polygon construction algorithm. Firstly, any 3D edge of a polyhedral object can be shared by at most two different planes. This if a segment of an evolving polygon chain incorporates a segment which has already been identified with two polygons then this line of search can be immediately abandoned. A second physical rule which is extremely useful is that the faces on opposite sides of a segment cannot lie in the same physical plane. Thus it is not possible for a partially developed polygon chain to have two consecutive segments shared with an already identified closed chain.

4.5 Results

In this section we show some results on real data to illustrate the performance of the polygon finder. The images were taken with a standard CCD camera and were processed on Sun-4 Sparcstation with code written in C. The input images were processed by a Canny edge detector and lines were identified in its thresholded output using a Hough-based line algorithm (Princen [60]). Figure 4-6 (a) and (b) show the gray-level image and subsequent lines extracted by the Hough process, respectively. It can be seen that edges do not always meet exactly at a junction. Figure 4-6 (b) shows the junctions used in the grouping process, where * is a V-junction and o is a Y or W-junction point. It is important
Feature Extraction

to note that there are many lines in the image which take part in junction formation but do not correspond to physically significant structures. These extraneous lines are due to effects such as shadowing. Also there is a lot of occlusion in the image caused by the juxtaposition of objects and self-occlusion. Thus the image is a difficult test for subsequent image interpretation.

In this example 48 V-junction were found and 20 Y- and W-junctions were identified. Following triple formation there were 111 triples. Figure 8 shows some of the polygons identified by the polygon finding process. Most of the structures in the image are triangles or quadrilaterals but the algorithm does successfully identify even the hexagonal shaped box. The program may identify, due to self and interobject occlusions, many sided polygons which are not physically interpretable as being due to single planar faces. However for this image only 2 such infeasible closed polygons extracted from the grouping process. In the case of open polygons, there are 15 open chains generated in the searching process. Some of these examples are shown in Figure 4-6(d). There are 6 open chains identified which are not part of a real object structure and there are 9 open chains which are part of a real object surface. After the generation of the V-junctions was completed, the generation of the triples, the searching for both closed and open polygons and the selection of the feasible polygons took 1.1 seconds. All single polygonal faces of the objects are found if the initial segmentation processes have correctly located lines that take part in junction formation.

4.6 Conclusions

In this chapter a novel polygon finding algorithm has been presented which works on real data and copes with many of the typical problems encountered in poorly segmented images. Both heuristic and physical rules have been utilised to control the search effort required in the polygon formation process. Experiments are demonstrated on real data and all polygons which could reasonably be due to single planar surfaces are found. Only a few spurious open polygon chains are identified.

The potential sources of knowledge which can be exploited in hypothesis generation and geometric matching. These connected triple and planar hypotheses will be used as an index into a database of models for objects which contain at least a triple of spatial edges or a spatial surface patch bounded by straight edges.
Figure 4-6: (a) A gray-level image of a polyhedral scene. (b) Line segments extracted by Hough process and junctions used in the grouping process. (c) The feasible closed polygons generated from the grouping process. (d) Examples of both the feasible and infeasible polygons extracted during the grouping process.
Chapter 5

Recognition of Polyhedral Objects Using Triangle-pair Features

5.1 Introduction

Model-based recognition of 3-D polyhedron under the perspective projection is an interesting and practically important topic in computer vision. Many man-made objects are polyhedral, especially industrial parts, buildings, furniture, etc. In the absence of 3-D sensors which in any case are either expensive, provide slow 3D-data acquisition or are difficult to calibrate, the interpretation of such objects may be based on the recognition, from their 2-D perspective projections, of the spatial planar polygons which constitute their surfaces.

In spite of the relatively simple form of polyhedral objects, their recognition has proved to be a very difficult problem. This situation is a consequence of the elusiveness of the solution to the problem of recognizing spatial planar polygons under perspective projection. Furthermore, an image usually contains data from many objects, as well as spurious and missing data caused by shadow, occlusion, surface markings and poor segmentation. Furthermore, the explicit depth information is not preserved during the process of projection. However, there are informative invariant 2D geometric features which can be extracted by initiating perceptual grouping operations that organise isolated low level primitives into larger scale structures conveying meaningful geometric cues. But in general such features are not powerful enough to resolve all the ambiguities inherent in a single perspective image. However, if complemented by high level, object specific constraints, both the robustness and efficiency of the matching process can be improved. Geometric models, in particular can provide highly constraining predictions for recognising well-defined object types such as polyhedrons.

In general, there are several distinct phases in model-base matching of rigid objects. Two off-line stages are model generation for constructing a CAD-like database of models, and model analysis for identifying and organising model features into structures for matching and for developing strategies for execution of the matching task. The two main run-time stages are hypothesis generation and
verification. The former consists of extracting interesting 2D geometric features from an image and then generating possible poses of scene objects so that the subsequent object verification process is provided with tight constraints on where to search for confirmatory evidence of model existence. The latter, model verification process performs a detailed check of the projected 3D features against 2D image data, confirming feature presence and accounting for features which are not observed. Most of the existing recognition systems using the above mentioned approach rely on geometric cues derived from the geometrical relationships between model-scene feature correspondences (Lowe [50], Horaud [34]).

In this chapter, we present a model-based recognition system (Wong [77]) for identifying the scene-to-model correspondences from a single perspective image. A hypothesize-verify paradigm based on local shape descriptions, namely triple-pair and quadrilateral features, is described. Geometric constraints derived from these key features are exploited to complete the battery of tools required to recognise general polyhedral objects.

Recently Lei [49] has developed a method for the recognition of spatial planar polygons under perspective projection based on a viewpoint invariant, namely, the cross ratio. However the approach is available for polygons of five sides or more only which restrict the type of polyhedron that can be recognised based on these features. Lowe [50] experimented with groups of powerful 2D non-accidental viewpoint independent cues in his SCERPO system to reduce substantially the number of inconsistent matches that may be considered in the matching process. Having selected a subset of informative features from the image, he proposed an iterative method to estimate the pose of the scene objects by refining the chosen initial transform parameters using a progressively greater number of hypothesized model-scene correspondences. However, the assignment of the initial values is a non-trivial task. Horaud [34] developed an effective hypothesis-verification scheme using triplet as a key feature. He provided a constructive method to recover the pose of a scene object using the geometric relationships between the corresponding model and scene vertices without any restriction on the angles between edges. Feasible solutions were searched for from a tessellated Gaussian sphere.

Thompson and Mundy [65] used a constraint propagation technique based on a Hough-clustering approach. In the framework, they have introduced a primitive feature, namely vertex-pair, formed by a pair of vertices connected by an imaginary line. Each of the model and scene vertex pair is used to compute a pose transformation that places the predicted model features in registration with the scene feature. Three degrees of freedom translation and three degrees of freedom rotation are embedded in a pose transformation. Thus each vertex-pair assignment can contribute a vote in a six-dimensional Hough space. Having considered all the model and scene vertex pair correspondences, a global peak is identified by searching through the six-dimensional Hough space. If the identified global peak is significant, this is indicative of the target object searched for being present in the scene. Their analysis of the vertex-pair, is based on affine transformation which is an orthographic projection with a scaling factor. This assumption is not acceptable in a general vision environment where the variation of the range of views is generally very large. Since our analysis of interesting
Recognition Using Triangle-pair Features

features, namely triangle-pairs, is based on perspective projection, the applicability of our method is not limited to images with weak perspective effect.

As mentioned earlier, our approach is based on triangle-pair features (Cheng [19]). The idea behind using triangle-pair features derives from the observation that spatial planar polygons are the constituent surfaces of polyhedral objects. Thus, they do not exist in isolation but rather are in specific geometric relations with each other. A quadrilateral feature can be analysed as a special case of a triple-pair feature. There are several reasons for employing these features as key features: the number of projections of triangle-pairs (called C-triple pairs) and quadrilateral features is generally manageable; they are qualitative viewpoint invariant geometric primitives. The transformation between a model and a camera frame can be completely determined using these features. The robustness of these features can be easily enhanced using an interactive environment between the matching phase and low level and feature grouping process.

To reduce the number of implausible hypotheses generated from scene-model triangle-pair and quadrilateral feature assignments, two effective geometric constraints, namely distance and angle constraints, are derived and incorporated in the matching process. Only those hypothesized model-to-scene feature correspondences which satisfy both the distance and angle constraints will be considered in the subsequent process.

After determining the pose of each hypothesis, we do not map the parameters of the hypothesized pose into a six-dimensional Hough space. Instead, the hypothesized objects are backprojected onto a 2D image plane. The predicted 2D features such as 2D line segments and angles between segments are used to correlate the predicted model with the features extracted from the scene. Having computed the confidence measure of each hypothesis, a global peak is searched for by scanning through the one dimensional confidence measure plot. If the identified peak is of significant height, this signifies that the target object is present in the scene.

Since the storage required for the plausible hypotheses after pruning is generally a small fraction of the accumulators required for the six-dimensional Hough space, the required storage of our approach will be commensurately lower than the one adopted in Thompson and Mundy's [65] work. Of course, some computational time must be devoted to the verification process, which will slightly erode the memory gains of our proposed approach. However, the verification stage in the matching process is quite simple and straightforward. Hence, the computational efficiency of our approach is higher than the Hough-clustering approach. Furthermore, the computation time required for searching for peaks in a single dimensional link list will certainly be lower than peak detection in a six-dimensional Hough space.

As a by-product of the matching process, the transformation defining the pose of the model with respect to the camera can easily be obtained using one of the two methods described in the paper. Our framework based on the hypothesize-verify technique using triangle-pair and quadrilateral features is very effective and intuitive. The derivation of the distance and angle constraints is simple. Furthermore, other interesting geometric primitives can be easily incorporated into our system to
Recognition Using Triangle-pair Features

handle more complex viewing environments.

The paper is organized as follows. In the next section, the solution of perspective equations for a spatial triangle is derived. The triangle recognition problem is considered in the context of candidate solutions obtained from the perspective equations. This will introduce an effective way to use distance and angle constraints to prune out implausible solutions and to identify a unique object from the model base which explains the observed data. Third, the distance and angle constraints are applied to the problem of detecting quadrilaterals. In Section 5.3, a method for estimating the scene object pose using the recovered triangle-pair or quadrilateral is described. In Section 5.4, the modules integrated into our polyhedral object recognition system are presented. Extensive experimental results obtained using the proposed method on three real images are presented in Section 5.5. Finally, conclusions are drawn about the proposed method.

5.2 The geometric constraints imposed by intermediate features

In this section, the triangle-pair (quadrilateral) primitive which can be used as a key feature for generating model hypotheses will be introduced. Two geometrical constraints imposed by this intermediate feature will be derived. The pruning power and reliability of the feature will be studied in an experiment. First, the analysis of the perspective projection of a spatial triangle is briefly overviewed in the next subsection.

5.2.1 A Perspective Projection of a Triangle

The difficulty in finding the correct model for a given perspective projection of a triangle is that different 3D triangular surfaces under perspective projection will produce the same 2D projected triangle. That is, in general all triangles in the database can be projected onto a single image triangle. One can always find a pose and an appropriate distance between the centre of the triangle and the image plane to achieve this.

If the corresponding points between a triangle and its projected image are known, then the perspective relationships between the model and its image can be computed easily. The details of the derivation and analysis of the perspective relationship can be found in (Fischler [28]). Here only a brief version of the derivation is outlined.

Suppose \( \triangle ABC \) in Figure 5-1 (a) is an object model and \( \triangle A'B'C' \) is its projected image. The point \( O \) is the origin of the camera coordinate system and \( O' \) is the origin of the image plane. Letting \( \alpha = \cos \angle A'OB', \beta = \cos \angle B'OC', \gamma = \cos \angle C'OA', k_x = k_x x \) and \( k_c = k_c y \). We have

\[
\begin{align*}
\alpha^2 &= k_x^2 (1 + x^2 - 2x\alpha) \\
b^2 &= k_x^2 (x^2 + y^2 - 2xy\beta) \\
c^2 &= k_x^2 (1 + y^2 - 2y\gamma)
\end{align*}
\]
where \( a, b \) and \( c \) are the lengths of the segments \( AB, BC \) and \( AC \) respectively, and \( k_a, k_b \) and \( k_c \) are the distances of vertices \( A, B \) and \( C \) of the triangle from the origin. Denoting \( n_1 = \left( \frac{a}{b} \right)^2 \) and \( n_2 = \left( \frac{a}{c} \right)^2 \), after some manipulation, the following biquadratic polynomial equation in one unknown \( x \) is derived,

\[
P_4x^4 + P_3x^3 + P_2x^2 + P_1x + P_0 = 0
\]  

(5.4)

where each coefficient \( P_i \) is a function of \( n_1, n_2, \alpha, \gamma \) and \( \beta \). The solution of Eq. (5.4) can be determined in closed form (Dehn [24]) or by iterative techniques (Conte [22]). In theory there are eight solutions for the Eqs. (5.1)-(5.3), since for every real positive solution there is a real negative solution [28]. In fact there are at most four possible solutions. For each positive real solution of Eq. (5.4), we can determine from Eq. (5.1) a single positive real value for the vertex distance \( k_a = \frac{a}{\sqrt{x^2 - 2x + 1}} \) and consequently \( k_b = k_c x \). From Eq. (5.3) on the other hand, we have

\[
y = \gamma \pm \sqrt{\gamma^2 + \left( \frac{\gamma^2 - \delta^2}{\delta^2} \right)}
\]

For each real positive value of \( y \) we obtain a value of \( k_c \) from \( k_c = k_a y \). Thus, it has been seen that the solution of \( k_a, k_b \) and \( k_c \) of the perspective equation with respect to the camera coordinate frame can easily be computed.

As the point correspondences are not normally known, there are up to a maximum of twelve possible solutions for distances \( k_a, k_b \) and \( k_c \) for any arbitrarily selected triangle model and a given 2D projected triangle extracted from a perspective image. Furthermore, there are corresponding values \( k_a, k_b \) and \( k_c \) for every constituent triangle surface of the polyhedral objects stored in the model base. Note that, a generic vision system would normally contain models of many objects. In view of these factors a detailed matching of each polyhedral object model against all the triangle features in order to obtain the best possible scene interpretation is therefore generally infeasible. Methods have to be developed for grouping isolated image primitives such as triangles into larger scale structures conveying meaningful geometric cues. In our framework, quadrilateral and triangle-pair structures are exploited as key features for model invocation and hypothesis generation. If the two hypothesized surfaces defining the constraint angle are not connected, the corresponding angle must be searched for by an exhaustive pairwise comparison between model surfaces of the hypothesized polyhedron. To reduce the search space, only the angles between touching model surfaces are used in the pruning process and not the spatial angles between pairs of non-connected faces.

The model-scene high-level feature assignments which satisfy these further constraints then become feasible candidates for consideration for the pose determination. The aim is for most model-scene assignments to be rapidly dismissed from consideration following a cursory examination using the constraints. In the next section, we will introduce the triangle-pair feature and its two geometric constraints will be derived.
5.2.2 The constraints : Triangle-pair

The proposed method exploits the fact that constituent surfaces of polyhedral objects are in specific geometric relations with each other. These relations can be used to resolve the inherently ambiguous interpretation of triangle-pairs which are encountered when triangular polyhedral object faces abut each other. We shall consider such a triangle-pair recognition problem first. Recall that the correspondence between the vertices of a model and of observed triangles is not known a priori. For example in Figure 5-1 (b) there are two possible potential matches which have to be considered between a triangle-pair model $\triangle ABC$ and $\triangle ACD$ and a triangle-pair image $\triangle A'B'C'$ and $\triangle AC'D'$. These are listed in Table A. To get a handle on the problem we shall invoke various constraints to reduce the solution set. In particularly we shall use higher level knowledge constraints. This in no way compromises our approach because the ultimate aim of identifying the model triangle from its projection is to recognise the polyhedron in which the triangle-pair constitutes one or part of its abutting faces.

The first constraint we adopt is a distance constraint. The basic principle of this approach is as follows. Suppose $A'B'C'D'$ is a projected line drawing of a polyhedron (see Figure 5-1 (b)). Obviously it consists of $\triangle A'B'C'$ and $\triangle A'C'D'$ abutting each other at the common edge $A'C'$. Note that the cardinality of the set of all the solutions $k_a, k_b, k_c, k_v$ of the perspective equation of the image $\triangle A'B'C'$ and $\triangle A'C'D'$ for each triangle-pair model in the database can be as many as 16. However, given that $A'C'$ is the projection of the common edge $AC$ between the model triangle-pair, only those solutions which satisfy $k_a = k_v$ and $k_c = k_v$, will be considered as plausible solutions. If the number of plausible solution is one, a unique model which corresponds to the feasible solution will be declared as the object in the scene. Usually using the distance constraint, the correct triangle model can be identified. But when the number of pairs of model triangles satisfying the distance constraint is greater than one, we must use an additional constraint to resolve the residual ambiguity. One such option is an angle constraint.

In Figure 5-1 (b), $\vec{P}$ and $\vec{Q}$ are the normals to $\triangle ABC$ and $\triangle ACD$, respectively. Here the positive direction of the normals is always pointed to the inside of the polyhedron. From Figure 5-1 (b) we denote,

\[
\frac{OA}{OA'} = \frac{k_a}{k_{a'}} = g; \quad \frac{OB}{OB'} = \frac{k_b}{k_{b'}} = h; \quad \frac{OC}{OC'} = \frac{k_c}{k_{c'}} = r; \quad \frac{OD}{OD'} = \frac{k_d}{k_{d'}} = s;
\]
Recognition Using Triangle-pair Features

then

\[
P = BC \times BA = \begin{vmatrix}
i & j & k \\
x'_c - hx'_b & y'_c - hy'_b & f(r - h) \\
x'_a - hx'_b & y'_a - hy'_b & f(g - h)
\end{vmatrix}
\] (5.5)

and

\[
Q = DA \times DC = \begin{vmatrix}
i & j & k \\
x'_a - sx'_d & y'_a - sy'_d & f(g - s) \\
x'_c - sx'_d & y'_c - sy'_d & f(r - s)
\end{vmatrix}
\]

where \((x'_i, y'_i)\) for \((i = a, b, c, d)\) is an image point with respect to the origin \(O'\) of the image plane and \(f\) denotes the focal length. The angle \(\varphi\) between \(\triangle ABC\) and \(\triangle ADC\) is

\[
\varphi = \pi - \cos^{-1} \left( \frac{\mathbf{P} \cdot \mathbf{Q}}{||\mathbf{P}|| ||\mathbf{Q}||} \right)
\] (5.6)

The measured angle \(\varphi\) should correspond to the actual angle between the faces of the hypothesized polyhedron. Since this information can be assumed to be known prior, the angle can be used to prune out the remaining inconsistent solutions.

Having discussed the distance and angle constraints, the integration of these constraints into a recognition framework results in the following procedure. First, all the models satisfying the distance constraints are selected from the model base for further consideration. If there is one such object model, we can then conclude that the object in the scene has been identified. Otherwise, the pre-computed angles between surfaces of the hypothesized models are compared with angle \(\varphi\) recovered from the scene. The object model satisfying the angle constraint will be associated with the scene object.

5.2.3 The Constraints: Quadrilateral

In this section, the technique of solving the 3-point perspective problem is used as a tool for identifying each quadrilateral in the scene. First note that each quadrilateral stored in the model base can be decomposed into two sets of triangular surface pairs using the diagonal of the quadrilateral as a common edge between them. For example, a quadrilateral \(ABCD\) shown in Figure 5-1 (c) is decomposed into two triangular surface pairs \(\triangle ABC\) and \(\triangle ACD\) taking the diagonal \(AC\) as a common edge and \(\triangle ABD\) and \(\triangle BDC\) taking the diagonal \(BD\) as a common edge. The four possible combinations between two sets of triangular surface pairs stored in the model and a triangle-pair taken from an image are listed in Table B. After selecting a correspondence between the model and the image, for example the first group in Table B, that is, \(\triangle ABC\) corresponding to \(\triangle A'B'C'\) and \(\triangle ACD\) corresponding to \(\triangle A'C'D'\), we solve the two perspective projection equations to obtain two groups of solutions. If the solutions for \(k_{a1}\) and \(k_{c1}\) under model \(\triangle ABC\) are equal to the solutions found for
Recognition Using Triangle-pair Features

Table B

<table>
<thead>
<tr>
<th>Scene</th>
<th>Triangle</th>
<th>Model</th>
<th>Triangle</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A'B'C'</td>
<td>A'B'C'</td>
<td>A'B'C'</td>
</tr>
<tr>
<td></td>
<td>A'B'C'</td>
<td>A'C'D'</td>
<td>A'B'C'</td>
</tr>
<tr>
<td></td>
<td>A'C'D'</td>
<td>A'B'C'</td>
<td>A'C'D'</td>
</tr>
<tr>
<td></td>
<td>A'B'C'</td>
<td>A'C'D'</td>
<td>A'B'C'</td>
</tr>
<tr>
<td></td>
<td>A'C'D'</td>
<td>B C D</td>
<td>B D A</td>
</tr>
</tbody>
</table>

$k_{a2}$ and $k_{c2}$ under model $\triangle ACD$ respectively, then we can conclude that quadrilateral $A'B'C'D'$ is possibly the projection of model $ABCD$.

If quadrilateral $ABCD$ is the true corresponding model, it must be a planar polygonal surface. Hence, if the triangular surface $ABC$ is part of the quadrilateral $ABCD$, the point $D$ of the triangle $BCD$ must lie on the plane of triangle $ABC$. This implies that the vector $\vec{CD}$ must be orthogonal to the normal $\vec{P}$ of the plane defined by triangle $ABC$. The formula for computing the normal $\vec{P}$ is given in Eq. (5.5) and the vector $\vec{CD} = (sx_d - rx_c)i + (sy_d - ry_c)j + f(s - r)k$. If the point $D$ is co-planar with $\triangle ABC$, then $\vec{P} \cdot \vec{CD} = 0$. Therefore the decision formula is

$$\begin{align*}
ss_d - rx_c & \quad sy_d - ry_c & \quad f(s - r) \\
rx_c - hx_b & \quad ry_c - hy_b & \quad f(r - h) \\
gx_c - hx_b & \quad gy_c - hy_b & \quad f(g - h)
\end{align*} = 0 \tag{5.7}
$$

The values of $g, h, r$ and $s$ computed using a triangular surface pair correspondence between models and the image, and the image points are substituted into Eq. (5.7) to check whether the quadrilateral model and the projected quadrilateral satisfy the constraint of coplanarity. The quadrilateral model which agrees with the coplanarity constraint will be registered as a consistent match.

5.3 Pose Estimation

As a by-product of the recognition method the corresponding vertices of the quadrilateral or the triangle-pair with respect to the camera frame are recovered. Using this information, the relative rotational parameters of the mapping from the model to its instance in the image can be computed. The transformation defines the pose of the model with respect to the camera. It can be obtained using one of two methods described in this section.

Consider the perspective geometry of the camera model depicted in Figure 5-1 (d). The image plane is assumed to be in front of the center of projection so as to acquire an upright image of the scene. The focal length, $f$ is the normal distance from the center of projection to the image plane. Based on the above configuration, the position of the scene vertex $P_s$ can be expressed in the camera frame $\mathcal{F}_C$ centered at the origin as $P_s = R_{MC} P_m + T_{MC}$, where $R_{MC}$ is the relative orientation between the model and camera frame and $T_{MC}$ is a translation vector. The corresponding model vertex $P_m$ is measured with respect to an model coordinate system $\mathcal{F}_M$.

To determine the relative rotation, we decompose the rotation transform into model-to-feature $R_{MF}$ and camera-to-feature $R_{CF}$ transforms. The transformation $R_{MF}$ of vertices $P_m$ expressed in the model coordinate system to vertices $P_v$ expressed in a feature-based coordinate system (see
Recognition Using Triangle-pair Features

Figure 5-1 (d) can be expressed as $P_v = R_{MF} P_m$, where $R_{MF} = \begin{pmatrix} m_{1x} & m_{1y} & m_{1z} \\ m_{2x} & m_{2y} & m_{2z} \\ m_{3x} & m_{3y} & m_{3z} \end{pmatrix}$ and $M_i = (m_{ix}, m_{iy}, m_{iz})$ is a unit vector. Likewise, the transformation $R_{GF}$ of vertices $P'$ defined in the camera coordinate system to vertices $P_v$ given in the feature-based coordinate system can be expressed as $P_v = R_{GF} P'_i$, where $R_{GF} = \begin{pmatrix} c_{1x} & c_{1y} & c_{1z} \\ c_{2x} & c_{2y} & c_{2z} \\ c_{3x} & c_{3y} & c_{3z} \end{pmatrix}$ and $C_i = (c_{ix}, c_{iy}, c_{iz})$ is a unit vector. Having determined $R_{MF}$ and $R_{GF}$, the rotation transform which maps an object model point $P_m$ to the corresponding scene feature point $P'_i$ in the camera coordinate system $F_C$ centered at the origin of the model frame $F_M$, can be written as $P'_i = R_{MO} P_m$ where $R_{MO} = R_{GF} \times R_{MF}$.

Now the meaning and methods of computing the unit vectors $M_i$ and $C_i$ for the case of triangle-pair and quadrilateral features will be discussed as follows. For this purpose, let us suppose the model-scene corresponding vertices $ABCD$ and $A'B'C'D'$ offer a plausible candidate. Consider the vectors $\vec{AB}$ and $\vec{AC}$ (see Figure 5-1 (c)), described with respect to a model coordinate system $F_M$. The orthogonal $x$-, $y$- and $z$-axis of a coordinate system basis can be constructed by

\begin{align*}
M_1 &= \frac{\vec{AC}}{||AC||} \\
M_2 &= \frac{\vec{BC} - (\vec{BC} \cdot M_1)M_1}{||\vec{BC} - (\vec{BC} \cdot M_1)M_1||} \\
M_3 &= \frac{M_1 \times M_2}{||M_1 \times M_2||}
\end{align*}

respectively. Likewise, the unit vectors $C_i$ can be determined from $A'B'$ and $A'C'$ using the same procedure. This method can be used for the case of a triangle-pair feature for computing the parameters of the rotational transformation by considering the vectors of any two edges of triangle $ABC$ or $ACD$ (see Figure 5-1 (b)).

5.4 Matching Strategy

5.4.1 Feature Extraction

In this section, we briefly present a simple algorithm for grouping 2D line segments into closed polygons and triangle-pair 2D features that correspond to plausible physical 3D structures of polyhedrons. The algorithm starts by identifying junctions created by two line segments whose end points are mutually proximal to within a junction threshold, measured along the line of intersection (Etemadi [26]). Having extracted the V-junctions from a scene, triples are formed by combining pairs of junctions which share a common line. These triples are then scanned by a procedure which connects them into polygonal structures. Heuristic rules are used to control the combinatorial explosion associated with
unconstrained association of junctions and triples. For additional details on the method of extracting closed polygons the reader is referred to chapter 4 (Wong [74]). In here, we shall describe the proposed method of identifying triangle-pair 2D features. The algorithm proceeds as follows.

- First, a list of triples, namely C-triple, is generated by merging pairs of V-junctions which share a common line segment, namely the intermediate segment. The resultant triples must have two end points located on the same half plane with respect to the intermediate segment.

- The aim of forming the triangle-pair 2D features is to find all C-triple pairs. To form a C-triple pair, the two merging C-triples must have a common segment. The intermediate segment of at least one of the two merging C-triples must be the common segment. Figure 5-2 (a) illustrates the case where the intermediate segments of the two merging C-triples form the common segment. In Figure 5-2(b), the intermediate segment of the C-triple (A) shares a common segment with one of the end segments of C-triple (B). Such cases will be identified as valid structures for further consideration in the grouping process to find triangle-pair 2D features.

- The result of the merging process is a list of C-triple pairs which are likely to be the projections of 3D triangle-pair features. These feasible candidates will be employed to provide a tight constraint on where to search for complementary V-junctions which yield plausible triangle-pair 2D features. In searching for significant triangle-pair 2D features, the algorithm attempts to identify V-junctions at the end points of the segments radiating from the common line segment of a C-triple pair. A triangle-pair is then identified if at least one V-junction is extracted from each of the two merging C-triples. For example, two plausible triangle-pairs 2D features \{e_1, e_2, e_3\} - \{e_1, e_5, e_7\} and \{e_1, e_4, e_5\} - \{e_1, e_6, e_7\} shown in Figure 5-2(c) are interpreted as 2D projections of a 3D triangle-pair structure.

Clearly, the robustness and reliability of the feature grouping depends entirely on the extraction of V-junctions and C-triples. In order to cope with inadequate low-level processing and partial occlusion, the poorer quality 2D junctions and C-triples can be accommodated by varying the threshold on the junction region size. However, extraneous and spurious closed loops and triangle-pair 2D features which do not arise from the projections of 3D local geometric shape may be extracted. Fortunately, the number of these features is generally manageable in the hypothesis generation and verification process and hence the computational cost is kept low. Many false hypotheses which are generated from these spurious features can be pruned away using the geometric constraints derived in this paper. Furthermore, the confidence measures computed by mapping the hypothesized models to the scene features using the transformation determined from the false model-scene assignments are relatively small and hence can be rejected. Having selected plausible geometric features from the line map, the method of generating feasible hypotheses by matching the extracted scene features against the model description will be described in the next section.
5.4.2 Hypothesis Generation

The role of this runtime stage is to identify model-scene feature assignments which satisfy the geometric constraints. The feasible model-scene feature assignments are used to estimate the pose of the target objects in the scene so that the subsequent verification process is provided with tight spatial constraints on where to search for confirmatory evidence of model instance. The procedures of this stage can be described as follows.

- The combinatorial assignments of the corresponding vertices between the triangle-pair and quadrilateral geometric features and the model description are considered. Using the model-scene triangle assignments, a list of possible solutions can be determined from the perspective Eq. (5.4). For each possible solution, the position and surface normal of the scene triangle measured with respect to the camera frame is computed.

- The model with at least one quadrilateral or triangle-pair satisfying the geometric constraints will be registered as a consistent interpretation. The relative rotation and translation embedded in the pose defining transformation are then estimated using the geometric relationships derived from the admissible model-scene assignment candidates.

Having determined the pose of each hypothesis, the method of computing the confidence measure by correlating the hypothesized model with the 2D scene features such as 2D junctions and straight line segments will be described in the next section.

5.4.3 Verification Process

The role of the verification process is to perform a detailed check of the correspondence between hypothesized object models and image data, confirming features present and accounting for features which are not observed. In this process, infeasible hypotheses are pruned away and the most plausible candidate is selected as an instance of the target object. The procedures exploited in the verification process can be described as follows.

- The 2D description of each hypothesized object is generated by backprojecting the model using the transformation computed in the hypothesis generation module. First, the hypothesized objects containing a hidden surface which is interpreted as a quadrilateral or a C-triple of a triangle-pair feature extracted from the scene are removed from the candidate list.

- For the remaining candidates we first count the number of 2D junctions of the hypothesized object that coincide with the junctions extracted from the scene image. Two junctions are said to coincide if they are within a proximity threshold and their angles and orientation must be within pre-specified allowable tolerances. After comparing every projected junction of the hypothesized object with the 2D junction extracted from the scene, the hypotheses with the
Recognition Using Triangle-pair Features

greatest number of matched junctions will be invoked for consideration in the next stage. The
aim of this stage is to select the hypothesis of greatest coincidence of features with the scene
data. To achieve this, the nearest scene line from each projected 2D model line is identified
within an allowable threshold and then each of the identified nearest scene lines is divided by
the corresponding projected line. These computed quotients are then summed up and divided
by the number of visible projected edges of the hypothesized model to yield a confidence
measure for the hypothesis.

After the confidence measure associated with each hypothesis is computed, a *global peak* is
searched for in the entire list of plausible hypotheses. If the confidence measure associated with
the identified global peak is significant the presence of an instance of the target object is confirmed.
Since, the global peak is identified from a confidence measure plot of the entire hypothesis list, this
signifies that our approach will not be prone to being trapped in local minima. If multiple target
objects are considered in the scene, the above mentioned matching process will be repeated for each
target object. Global peaks are searched for from each confidence measure plot generated for each
hypothesized model. The performance of the matching system will be validated in the experiments.

5.5 Experimental Results

In this section, some experimental results are presented to demonstrate the effectiveness and robustness
of the matching strategy incorporated into our recognition system. All the test images are taken with
a standard CCD camera and were processed on Sun-4 Sparcstation with code written in C language.
Real images containing polyhedral objects were employed to test the efficacy of our method. An
object-centred and viewer-centred right-handed coordinate systems were used to define the vertices
of object models and scene objects respectively.

The first experiment involved 4 polyhedral object models of different shape. The test image
contained the target objects being placed in random orientation without occlusion. The aim of this
experiment was to test the effectiveness and reliability of the two high-level geometric features
employed as seed features for generating feasible hypotheses. The vertices of the polyhedral object
models are shown in Figure 5-3. The test image shown in Figure 5-5 (a) was processed by a Canny
dge detector, and straight line segments shown in Figure 5-5 (b) were extracted from its thresholded
output using a Hough-Based line algorithm. It can be seen that edges do not always meet exactly
at a junction. Also there are many lines extracted from the scene which participate in the high-level
feature grouping process even though they do not correspond to physical 3D edge structures.

For this test image, 4 quadrilaterals (C1,...,C4) and 22 triangle-pair features (F1,...,F22) were
found. These are tabulated in Table C of Figure 5-4 along with the line segments forming these
features. The C-triple notation ( e1, e2, [ e3, e4 ] ) indicates a common junction has been identified
as the intersection of the edges ( e2, e3 ) or ( e2, e4 ). Closed chains of more than 4 sides extracted
by the grouping process were not considered. Five triangle-pair features F4, F5, F9, F10 and F14
which were produced during hypothesis generation do not correspond to the projection of a 3D triangle-pair feature. It is worth noting that both the true and spurious features are employed as key features for generating feasible hypotheses in the matching process.

All the high-level features extracted from the scene were matched exhaustively against the pre-processed description of the 3D models. Column [t] of Table D of Figure 5-4 lists the number of feasible model-to-scene feature assignments extracted before applying geometric constraints. As can be seen from the table, the number of solutions can be very large. In all cases, there were feasible solutions found when establishing the perspective geometric relationships between the individual triangular features of the models and the test scene. The total number of feasible solutions determined from matching the triangular prism #1, roof model #2 and box model #3 against all the quadrilateral features \(C_1, \ldots, C_4\) and triangle-pair features \(F_1, \ldots, F_{22}\) was 3438, 3492 and 4862 respectively (see bottom row of Table D of Figure 5-4). As the pyramid model #4 does not contain surfaces made up from 4 sides, the matching of the quadrilateral features against the pyramid model was not necessary. In the case of matching the pyramid model #4 against the triangle-pair features \(F_1, \ldots, F_{22}\), the number of feasible solution was 1114. The admissible solutions were then checked using the distance constraint derived from the high-level features.

In the experiment, a reasonably large error in matching the endpoint positions of the common line segments shared by the two triangle features was allowed. This was to prevent the 'pruning out' of true high-level features due to inadequate low level processing. After applying the distance constraint, the number of feasible candidates for the models #1, #2, #3 and #4 was reduced by 56.2 %, 65.8 %, 46.7 % and 64.8 % i.e. about half the hypotheses were rejected. The established match of models against each extracted scene feature is shown in column [d] of Table D of Figure 5-4.

The admissible candidates satisfying the distance constraint were then tested with the angle constraint. In general, we chose a large allowable tolerance when comparing the corresponding angles between the abutting model and scene surfaces. In this experiment, the tolerance was chosen to be 20°. In allowing such a high tolerance, false hypotheses will necessarily be retained in the list and passed to the verification phase which will slightly reduce the overall efficiency of the recognition process. However, the probability of rejecting the correct interpretation is greatly reduced. After applying the angle constraint, the proportion of geometrically admissible candidates to feasible solutions without applying the geometric constraints for the models #1, #2, #3 and #4 reduces to 10.4 %, 7.4 %, 17.2 % and 8.6 %, respectively. A summary of the obtained match of models against each scene feature using both the distance and angle constraints is shown in column [a] of Table D of Figure 5-4. The number of implausible hypotheses has been pruned down dramatically.

Some examples of incorrect and poor quality hypotheses generated by matching the box model #3 against feature \(F_1, F_6\) and \(F_{18}\) are labelled \(H_1, H_2\) and \(H_3\) respectively (see Figure 5-5(c)). The perspective distortion of the hypothesized backprojected model \(H_2\) shown in Figure 5-5(c) was quite severe. Representative hypotheses generated by matching the triangular prism model #1 against the scene features \(F_{14}\) and \(F_6\) are shown in Figure 5-5(d) and marked with \(H_1\) and \(H_2\) respectively.
Recognition Using Triangle-pair Features

is worth noting that the scene feature F14 taking part in the matching process was produced by two extraneous line segments due to effects such as shadow. Some hypotheses generated by matching roof model #2 against the scene feature F8, F9 and F19 are labelled H1, H2 and H3 respectively (see Figure 5-5(e)). The pose of these hypothesized models was very unstable with respect to the supporting surface. Representative incorrect hypotheses arising from matching the pyramid model #4 against features F6, F8, F4 and F18 are shown in Figure 5-5(f) and labelled with H1, H2, H3 and H4 respectively.

The confidence measure of each admissible hypothesis was computed. The highest confidence measure is tabulated in column [q] of Table D of Figure 5-4. At this point, the highest confidence measure of each object model was identified. The hypotheses generated by matching the models to the correct scene features are shown in Figure 5-5(g) and (h). The hypotheses generated by matching the triangular prism #1 against the scene features F12 and F8 yielding the highest confidence measures are shown in Figure 5-5(g) marked S1. All the hypotheses were very close to the correct one and their confidence measures are quite significant (all above 81%). Another correct scene feature C1 yielded a relatively high confidence measure (89.0%) when matching against the triangular prism #1. The plausible hypotheses generated by matching the roof model #2 against the scene feature F6, and which yield the highest correspondence measure, are labelled S2 and depicted in Figure 5-5(g). The translation vectors of two of these hypotheses were very different from the true one. Fortunately, these hypotheses yield a very low confidence measure (about 32%). The hypotheses S3 and S4 generated when matching the pyramid model #4 and box #3 against the feature F1 and C3 are shown in Figure 5-5(i). Some of the hypothesized box models S4 are suspended above the supporting surface.

In this experiment, the triangular prism #1, roof model #2, box model #3 and pyramid model #4 matched the scene object #1, #3, #4 and #2 correctly. The confidence measure in each of these cases is 90.9%, 72.1%, 85.7% and 90.8% respectively. After the extraction of high-level features was completed, the hypothesis generation and verification process took 2 minutes and 47 seconds for identifying the correct solution. The residual ambiguous hypotheses did not affect the reliability and computational efficiency of the recognition system. Most of these false hypotheses were associated with relatively low confidence measures and could easily be rejected on that basis.

The second experiment involves the same polyhedral object models as in the previous experiment. The test scene contained three target objects surrounded and occluded by other objects (see Figure 5-8(a)). The triangular prism and roof model were placed in an unstable position. The aim of this experiment was to study the computational feasibility and robustness of the proposed method of object recognition in the presence of clutter and subject to partial occlusion. The output of the low level processing contained many spurious line segments which were due to effects such as shadowing (see Figure 5-8(b)). Also there were some V-junctions in the image caused by self occlusion. In this example, 2 quadrilaterals (C1, C2) and 8 triangle-pair features (F1, ..., F8) were identified and tabulated in Table E of Figure 5-6. The labels of the line segments forming these features are tabulated in
Recognition Using Triangle-pair Features

Table E of Figure 5-6. Only three of all these extracted features correspond to any 3D physical structure of the object models. Thus this image is a difficult test for any interpretation procedure.

All the high-level features were exploited as seed features and were matched exhaustively against the 4 models shown in Figure 5-3. Based on the perspective projection analysis, the total number of feasible solutions determined for the triangular prism #1, roof model #2, box model #3 and pyramid model #4 was 1344, 1348, 1896 and 412, respectively. After applying the distance constraint, the number of geometrically admissible solutions in each case was reduced by 68.2%, 75.8%, 61.2% and 79.1%, respectively. The residual candidates were then checked with the angle constraint. After performing the test, the proportion of remaining admissible candidates dropped to 8.3%, 7.1%, 8.3% and 2.9%. These results illustrate the powerful role of the distance and angle constraints.

Representative hypotheses generated by matching the triangular prism #1, roof model #2, box model #3 and pyramid model #4 are shown in Figure 5-8 (c), (d), (e) and (f) respectively. Two hypotheses generated by matching the base of triangular prism #1 and roof model #2 against the projected planar face $C_1$ of the small cube are shown in Figure 5-8 (c) and (d) and labeled with $H$. Significantly, the shortest distance computed for the two hypothesized models, with respect to the camera frame, differed from the true value by a factor of five. The confidence measures in matching of the scene features against the pyramid model #4 were relatively weak (all below 40%). Hence, we can declare that the pyramid model is not present in the scene. Some pyramid hypotheses of confidence measures above 30% are shown in Figure 5-8(f).

The confidence measure of each geometrically admissible candidate was computed. The highest confidence measure obtained when matching an object model against each extracted feature is tabulated in column $[q]$ of Table F of Figure 5-6. The hypothesized object model yielding the highest confidence measure was determined. The hypotheses generated by matching the object models to the correct high-level features are shown in Figure 5-8(g). The hypotheses generated by matching the triangular prism #1 and roof model #2 against the scene features $F_3$ and $F_2$ yielding the highest confidence measures are shown in Figure 5-8(g) marked $S_2$ and $S_1$ respectively. In this experiment, the triangular prism #1 and roof model #2 matched the scene object #2 and #1 correctly. The confidence measure in each of these cases is 76.8% and 78.1% respectively. When computing the confidence measures for the hypotheses generated for the box model, the most plausible but incorrect candidate, among the hypotheses generated by matching against scene feature $C_2$, gave a confidence measure 2.3% lower than the best four hypotheses (55.3%) generated from matching the model against the scene feature $C_1$. This wrong interpretation (shown in Figure 5-8(e) labelled $H$) was due to the fact that the quality of the 2D line description of the target box model extracted from the scene was slightly poorer than the small cube. This was because the target box model was partially occluded by the pyramid model, hiding some of the significant local geometric 3D structures. Furthermore, the 2D description of the small cube and target box were very similar in terms of 2D shape under perspective projection. As a result, the assignments between the box model and the 2D features of the small cube will always be geometrically admissible. However, the correct hypothesis can be found.
if a bound on the distance from the camera to the target object is known a priori. The computed shortest distance for the four hypotheses generated by the small cube at the top of the list differed from the edge of the supporting surface by a factor of four. Hence, these invalid hypotheses can be pruned away from the list. In practise, the bounded area of the supporting surface may not be generally known in advance. However, the box model could only be identified from the scene by making this weak assumption. The hypotheses generated by matching the box model #3 against the correct scene feature C2 is shown in Figure 5-8(g) and labelled with S3. The models superimposed on the scene image using the computed transformation are shown in Figure 5-8(h).

Next, the proposed matching strategy was tested on a cluttered scene consisting of a ECB model (electrical circuit breaker) mounted on the wall in our vision laboratory (see Figure 5-9 (a)). The target ECB model was surrounded by several arbitrary shape features. A simplified version of the ECB model #5 is shown in Figure 5-3. In this experiment, all the object models shown in Figure 5-3 were involved in the matching process. The aim of this experiment was to study the effectiveness and capability of the system in recognising model objects in a complex environment. Figure 5-9 (b) shows the output of the Hough based line finding process.

In this test image, 4 quadrilaterals (C1,...,C4) and 6 triangle-pairs (F1,...,F7) were identified from the scene using the feature grouping process. The labels of the line segments forming these features are tabulated in Table G of Figure 5-7. The quadrilateral features C1 and C2 were generated by the 'instructions face' of the ECB model and an adhesive memo paper stuck on the wall, respectively. It is worth noting that the scene feature C4 was not extracted from a planar surface of the ECB model, as the edge index 13 was extracted from the edge of an upright surface of the ECB model. In fact, there was only one relevant quadrilateral feature extracted from the scene. Three of the triangle-pair features F5,F6 and F7 were generated by noise. One triangle-pair feature formed by edges (19, 20, 24) and (19, 9, 29) was removed from the list of candidates as the edge 9 was parallel to 19 which does not correspond to the physical 3D structure in general view.

The preprocessed 3D descriptions of the models were matched exhaustively against the high-level features extracted from the scene. The perspective analysis of quadrilateral and triangle-pair structures precomputed from the model and the scene features was performed. Before applying the distance and angle constraints, the number of feasible solutions determined for the triangular prism #1, roof model #2, box model #3 and pyramid model #4 and ECB models #5 was 1144, 1094, 1612, 288 and 1886, respectively. After applying the distance constraint, the number of admissible candidates in each case was reduced by 68.0 %, 67.7 %, 56.9 %, 67.2 % and 72.6 %.

The confidence measure of each admissible hypothesis listed in column (a) of Table H of Figure 5-
Recognition Using Triangle-pair Features

7 was computed. The corresponding highest confidence measure is listed in column \([q]\) of Table H of Figure 5-7. Representative hypotheses generated by matching the triangular prism \#1, roof model \#2, box model \#3, pyramid model \#4 and ECB model \#5 are shown in Figure 5-9 (c), (d), (e), (f) and (g) respectively. It is interesting to note that the scene feature \(F7\) participating in the matching process was produced by spurious line segments due to effects such as shadow. Some hypotheses generated by matching triangular prism \#1, pyramid model \#4 and ECB model \#5 are shown in Figure 5-9 (c), (f) and (g) marked \(H\), respectively.

The hypothesized object model yielding the highest confidence measure of sufficient quality was interpreted as an instance of the object in the scene. Computing the confidence measures for the matches of the ECB model against all the scene features (see column \([q]\) of Table H of Figure 5-7), the highest confidence measure (68.2\% ) was observed for the correct scene feature \(F2\). As the confidence measures in matching of scene feature \(F2\) against the other four object models were relatively low (all below 50\%), the correct model for the scene object was identified.

The hypotheses generated by matching the ECB model \#5 against the correct scene feature \(F2\) are shown in Figure 5-9(h) and labelled with \(S\). One of the hypotheses was suspended in space. The models superimposed on the scene image using the computed transformations are shown in Figure 5-9(i).

It is important to note that the highest confidence measure computed when matching the roof model and the scene feature \(C1\) was very high (94.8\%). This is due to the fact that the rectangular base of the roof model under perspective analysis is similar to the scene feature \(C1\) of a rectangular shape (see Figure 5-9(d) marked \(W\)). Furthermore, the only feature involved in computing the confidence measure was the visible rectangular surface of the transformed roof model in an accidental view.

5.6 Conclusions

In this chapter, we have presented a hypothesize-verify approach to polyhedral recognition based on the use of geometric constraints derived from local shape properties. In the framework, two intermediate features, namely triangle-pair and quadrilateral are employed as key shape descriptors for identifying a manageable number of geometrically feasible model-to-scene candidates. Two effective geometric constraints, namely the distance and angle constraint, have been derived and incorporated into our recognition system. Many infeasible hypotheses can be swiftly pruned away from the hypothesis list. Only those hypothesized model-to-scene correspondences which satisfy the distance and angle constraints are considered in the subsequent (computationally intensive) verification process. The integration of the feature extraction, hypothesis generation and verification process is described. Extensive experimental results using real images have been presented. They verify the effectiveness and reliability of our proposed method. As a by-product of the matching process, the transformation defining the pose of the scene objects with respect to the camera has been recovered. The experimental results show that the localization method has a reasonable accuracy in estimating
the pose of the target object. The capability of the proposed recognition system was tested on many real images (≈ 80). In general, the success rate for identifying and localising target objects using the proposed paradigm is relatively high. It will occasionally give false results when the projection of the target object is relatively small compared to the size of the image plane.

Clearly, one of the prerequisites of the proposed method is the ability to extract triangle-pair and quadrilateral features from low level primitives such as junctions from image data. To accommodate the problems of noise, oversegmentation or undersegmentation, the poorer quality junctions will participate in the feature extraction process by allowing large thresholds on proximity and orientation checks. As a result, both the number of plausible and implausible features included in the matching process will be increased. However, the probability of rejecting the correct model-to-scene feature correspondences will be greatly reduced. Moreover, the growth of hypotheses generated from the model-to-scene assignments will not affect the computational performance of the system, as a majority of these hypotheses which are geometrically infeasible will be pruned away.

Most of the existing verification processes are based on identifying correspondances among image features for hypothesised low level primitives whose positions are computed from the estimated 3D position of the target object. The performance of this approach will degrade significantly when verifying model hypotheses generated in a complex scene. To increase the robustness of the verification process, the use of spatial and temporal context will be considered in future. The spatial description of a scene can be constructed by aggregating matching results derived over a period of interest. For example, a box model was interpreted as a small cube in the scene. In this instance, if the supporting surface such as a table is identified and maintained in the global scene description, the incorrect interpretation could then be rejected using the bounded space of the table.
Recognition Using Triangle-pair Features

(a) A perspective projection of a triangle

(b) Constraints of a triangle-pair feature

(c) Constraints of a quadrilateral feature

(d) A simplified pin-hole camera model

Figure 5-1: See text
Recognition Using Triangle-pair Features

Figure 5-2: Triangle-pair features

Figure 5-3: Object Models
<table>
<thead>
<tr>
<th>Index</th>
<th>Edge labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>(28, 31, 32, 30)</td>
</tr>
<tr>
<td>C2</td>
<td>(1, 2, 8, 7)</td>
</tr>
<tr>
<td>C3</td>
<td>(2, 3, 4, 5)</td>
</tr>
<tr>
<td>C4</td>
<td>(1, 6, 3)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table C</th>
<th>Table D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>triangular prism #1</td>
</tr>
<tr>
<td></td>
<td>t</td>
</tr>
<tr>
<td>Cl</td>
<td>54</td>
</tr>
<tr>
<td>C2</td>
<td>48</td>
</tr>
<tr>
<td>C3</td>
<td>56</td>
</tr>
<tr>
<td>C4</td>
<td>48</td>
</tr>
</tbody>
</table>

| F1     | (10, 14, 15, 16) | (10, 13, 12, 18) |
| F2     | (3, 2, 15, 16) | (3, 9, 6) |
| F3     | (3, 2, 15, 16) | (3, 1, 6, 7) |
| F4     | (10, 14, 15, 16) | (10, 13, 12, 18) |
| F5     | (10, 13, 12, 18) | (10, 19, 18) |
| F6     | (27, 26, 24) | (27, 26, 24) |
| F7     | (30, 28, 31) | (30, 28, 31) |
| F8     | (27, 23, 21) | (27, 26, 24) |
| F9     | (27, 23, 21) | (27, 26, 24) |
| F10    | (27, 23, 21) | (27, 26, 24) |
| F11    | (3, 4, 5) | (3, 1, 6, 7, 1) |
| F12    | (3, 25, 29) | (30, 20, 31) |
| F13    | (3, 4, 5) | (3, 9, 6) |
| F14    | (13, 15, 16) | (12, 19, 18) |
| F15    | (3, 4, 7) | (3, 8, 7) |
| F16    | (1, 6, 9) | (1, 7, 8) |
| F17    | (2, 3, 4, 5) | (2, 3, 4, 5) |
| F18    | (1, 6, 9) | (1, 7, 8) |
| F19    | (2, 3, 4, 5) | (2, 3, 4, 5) |
| F20    | (1, 6, 9) | (1, 7, 8) |
| F21    | (2, 3, 4, 5) | (2, 3, 4, 5) |
| F22    | (1, 6, 9) | (1, 7, 8) |

| Total | 1597 | 357 | - | 1542 | 1134 | 258 | - | 2492 | 2392 | 857 | - | 1114 | 592 | 96 | - | - |

Figure S-6: The detailed matching results
Recognition Using Triangle-pair Features

(a) A test image
(b) A line drawing
(c) Box model hypotheses
(d) Triangular prism hypotheses
(e) Roof model hypotheses
(f) Pyramid hypotheses
(g) see text
(h) see text
(i) The matching results

Figure 5-5: The scene image and the experimental results
### Table E

<table>
<thead>
<tr>
<th>Index</th>
<th>Edge labels</th>
<th>Triangular prism #1</th>
<th>Roof model #2</th>
<th>Box model #3</th>
<th>Pyramid model #4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>t d a q</td>
<td>t d a q</td>
<td>t d a q</td>
<td>t d a q</td>
</tr>
<tr>
<td>C1</td>
<td>(1, 3, 24, 12)</td>
<td>52 46 13 38.3</td>
<td>56 16 2 38.9</td>
<td>104 96 23</td>
<td>55.3</td>
</tr>
<tr>
<td>C2</td>
<td>(20, 23, 22, 21)</td>
<td>56 42 16 61.7</td>
<td>48 22 16 51.4</td>
<td>96 96 55</td>
<td>53.0</td>
</tr>
<tr>
<td>F1</td>
<td>(4, 7, 50)</td>
<td>156 36 6 17.0</td>
<td>156 12 0</td>
<td>224 8 0</td>
<td>48 0 0</td>
</tr>
<tr>
<td>F2</td>
<td>(11, 10, 29)</td>
<td>152 58 8 53.3</td>
<td>152 46 8 78.1</td>
<td>208 80 40</td>
<td>32.7</td>
</tr>
<tr>
<td>F3</td>
<td>(17, 16, 18, 17)</td>
<td>160 70 14 75.8</td>
<td>160 36 4 22.0</td>
<td>192 16 0</td>
<td>64 22 0</td>
</tr>
<tr>
<td>F4</td>
<td>(17, 16, 18, 17)</td>
<td>144 36 16 69.8</td>
<td>140 20 8 46.6</td>
<td>192 32 16</td>
<td>38.4</td>
</tr>
<tr>
<td>F5</td>
<td>(23, 21, 30, 29)</td>
<td>156 20 4 35.0</td>
<td>100 34 12 43.2</td>
<td>240 128 16</td>
<td>25.6</td>
</tr>
<tr>
<td>F6</td>
<td>(23, 21, 22, 23)</td>
<td>156 38 12 41.7</td>
<td>156 34 16 43.9</td>
<td>240 120 0</td>
<td>- 50 4 0</td>
</tr>
<tr>
<td>F7</td>
<td>(2, 13, 23)</td>
<td>156 54 6 52.3</td>
<td>156 76 20 39.2</td>
<td>208 132 8</td>
<td>22.1</td>
</tr>
<tr>
<td>F8</td>
<td>(17, 16, 18, 19)</td>
<td>144 28 16 69.8</td>
<td>140 20 10 46.6</td>
<td>192 8 0</td>
<td>- 48 0 0</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>1344 428 111</td>
<td>- 1348 326 96</td>
<td>- 1896 736 158</td>
<td>- 412 86 12</td>
</tr>
</tbody>
</table>

**Figure 5-6:** The detailed matching results

### Table G

<table>
<thead>
<tr>
<th>Index</th>
<th>Edge labels</th>
<th>Tri prism</th>
<th>Roof model</th>
<th>Box model</th>
<th>Pyramid model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>t d a q</td>
<td>t d a q</td>
<td>t d a q</td>
<td>t d a q</td>
</tr>
<tr>
<td>C1</td>
<td>(1, 6, 5, 2)</td>
<td>54 20 61.6</td>
<td>60 8 54.8</td>
<td>120 44 62.9</td>
<td>-</td>
</tr>
<tr>
<td>C2</td>
<td>(11, 14, 15, 10)</td>
<td>56 6 30.6</td>
<td>50 3 60.8</td>
<td>146 11 36.0</td>
<td>-</td>
</tr>
<tr>
<td>C3</td>
<td>(7, 28, 3, 27)</td>
<td>56 18 52.0</td>
<td>48 10 52.0</td>
<td>96 44 58.3</td>
<td>-</td>
</tr>
<tr>
<td>C4</td>
<td>(7, 28, 13, 27)</td>
<td>56 14 51.9</td>
<td>56 13 70.9</td>
<td>96 47 53.9</td>
<td>-</td>
</tr>
<tr>
<td>F1</td>
<td>(28, 23, 22, 11)</td>
<td>156 0 - 148 10</td>
<td>- 191 32 35.1</td>
<td>48 1 48.1</td>
<td>240 8 66.0</td>
</tr>
<tr>
<td>F2</td>
<td>(28, 7, 57,264) (28, 28, 16, 15, 30)</td>
<td>144 4 12.9</td>
<td>148 4 30.2</td>
<td>192 24 33.3</td>
<td>48 2 37.0</td>
</tr>
<tr>
<td>F3</td>
<td>(28, 28, 14,24) (7, 27, 13, 18)</td>
<td>156 12 45.2</td>
<td>144 12 44.2</td>
<td>151 16 49.2</td>
<td>48 3 54.8</td>
</tr>
<tr>
<td>F4</td>
<td>(7, 20, 16, 23), (7, 28, 13, 19)</td>
<td>156 2 15.0</td>
<td>144 12 44.2</td>
<td>192 8 46.9</td>
<td>48 0 - 226 6 67.1</td>
</tr>
<tr>
<td>F5</td>
<td>(7, 26, 13, 17, 7, 28, 13)</td>
<td>156 0 - 148 16 43.3</td>
<td>192 8 46.9</td>
<td>48 0 - 226 0 -</td>
<td></td>
</tr>
<tr>
<td>F6</td>
<td>(19, 8, 3, 21, 24)</td>
<td>156 8 41.1</td>
<td>148 0 - 224 16 27.4</td>
<td>48 5 40.9</td>
<td>232 8 23.4</td>
</tr>
<tr>
<td>F7</td>
<td>(31, 18, 17, 21, 12, 38, 3)</td>
<td>216 10 54.8</td>
<td>216 4 - 288 0 - 72 1 44.3</td>
<td>352 10 33.8</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>1368 94 - 1318 92 - 1902 250 - 360 42 - 2226 227 -</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 5-7:** The detailed matching results
Recognition Using Triangle-pair Features

Figure 5-8: The scene image and the experimental results
Recognition Using Triangle-pair Features

Figure 5-9: The scene image and the experimental results
Chapter 6

Analysis of a triple of spatial edges

6.1 Introduction

This chapter introduces a new approach to analysing a triple of spatial edges and its corresponding triple of image lines obtained from their single perspective view. No restriction is imposed on the configuration of the triple edges. Furthermore, our approach is line-based and hence is not dependant on end points of image lines being detected. Therefore, compared to point-based, this methods will be more reliable and robust in the presence of noise. The results derived here can be widely used in various computer vision paradigms.

An intuitive mathematical formulation is proposed to interpret the geometric relationships between a triple of spatial edges and their perspective projection forming image lines. To analyse geometric features under perspective projection, Kanatani has developed a technique to move a scene feature, the focus of interest, to a canonical position where the perspective analysis can be greatly simplified. Using this elegant technique we derive an eight degree polynomial for interpreting a triple of space lines in general configuration under perspective projection. Particular closed form solutions are derived for specific but common configurations of edges such as rectangular bar end and orthogonal triple. Although our framework is inspired by Kanatani [44], the extension of his technique is far from obvious. To date, the author believes that the proposed method for analysing triple spatial edges under perspective view is completely new. Furthermore, the practical significance and generality of the results are multifold. This will become apparent from the later sections and from the experimental results.

Having established the basic formulation for interpreting a triple of spatial edges from their corresponding triple of image lines, a model-based polyhedral object recognition system for identifying the scene-model correspondences and estimating the pose of scene objects from a single perspective image has been developed. A recognition system uses a hypothesis-verification paradigm based on local shape properties. In the framework, a triple of connected spatial edges and a trihedral vertex are employed as key features for model invocation and hypothesis generation. A trihedral vertex is a 3D spatial vertex with three space lines radiating from the same tip. The geometric constraints of
these key features are the basis for their implementation. There are several reasons for choosing the triple: the number of such features extracted from a scene is generally manageable; they are robust in the presence of moderate noise; they are qualitative invariants over a wide range of view points [42]; they can constraint the transformation between the model and camera frames.

Given a triple of image lines, the derived eight-degree polynomial or geometric constraint equation is explicitly defined by the space angles between the corresponding three edges measured with respect to an object centered coordinate system. The crux of this representation is that the angular attributes of pairs of spatial edges are object-independent. This means that for a given image triple, all triples of space lines sharing the same angle sets will be bounded by the same geometric constraint equation. Because of this representation, effectiveness of a proposed constraint equation generation scheme can be derived by taking the advantage of the commonality between models, in this case the space angles between edges, to avoid replicating the similar module in deriving and solving the geometric constraint equation. This can be easily achieved by classifying the model triple features of common space angles into the same triple group.

The bulk of computation involved in deriving and solving the geometric constraint equation only needs to be performed once for a given image and model triple correspondence. The computational load depends on the number of distinct groups of triples. For this work, groups are distinguished by the angles between the constituent model edges. Many objects will contain triples which fall into existing groups and this implies there will be no increase in computation when they are added to the model database. Hence computation can be sublinear in terms of the number of models in the list.

Furthermore, if no real solutions are found in solving the constraint equation, the group of triples sharing the same common space angles will not be considered in the subsequent pose determination module. Hence, a large number of model and image triple correspondences can be pruned away at the early stages of the hypothesis generation phase. Hence, the overall efficiency of the recognition system based on this scheme will increase significantly. This however is only true, if the model triples share many common space angles. This empirical observation is quite reasonable, especially in the case of a large model base containing many polyhedral objects.

Once the real roots of the constraint equation have been determined, the orientations of the triples of space lines measured w.r.t the camera frame can be easily computed. At this stage, it is important to note that scene knowledge has been recovered using relative angular information, it does not use any model information described in the object centered coordinate system such as the vector equation of a model edge. Once the edge orientations are recovered, the relative rotation and translation transforms between the model and camera frames can be established. Finally a verification process is employed to perform a detailed check of the correspondence between predicted projected lines and extracted image lines.

Although, the recognition system described in this chapter is solely based on a hypothesis-verification paradigm, we explore the feasibility of these features in a recognition system based
Analysis of a triple of spatial edges

on a Hough-clustering approach. From the experimental results, we verify that triple features are plausible candidates use in conjunction with the clustering approach.

The main contributions of this chapter include the following.

- A new and intuitive formulation of the problem of interpreting a triple of spatial edges and their corresponding perspective image lines is introduced and analysed. Closed form solutions are derived for three special and common configurations, namely, generic rectangular bar, orthogonal and trihedral vertex.

- The geometric constraint equation for a scene triple is derived and established in an object independent coordinate frame. Given the description of an image triple, the constraint equation is explicitly defined by the 3D space angles between the corresponding model triple edges. As, the constraint equation is derived using the angular relationship between edges, it does not require any metric information measured with respect to an object coordinate frame.

- The geometric constraint equation will be derived and solved only once for a given image triple and the common space angles shared by a group of connected model triples. This does not depend on the number of model triples being classified in the triple group.

- In the framework of a model-based polyhedron recognition system, we have proposed to use triples of connected edges as key features. These features are computationally feasible, have local support and are stable in the presence of moderate noise. During the model organisation phase, the space angles between the connected edges of a model triple can be easily computed. Connected triples with the same space angles can be classified into a common group to take advantages of the above constraint equation generation scheme.

- Extensive experiments have been performed to verify the plausibility for employing connected triple edges and trihedral vertices as key features in the paradigm of hypothesis-generation and Hough-clustering approaches to object recognition.

6.2 Related Work

Shakunaga and Kaneko [64] have developed a general method for analysing a triplet of spatial edges. They have proposed a complicated mathematical framework for solving this problem. An analytical solution is given for the case where one of the space lines is orthogonal to the other two. In the general case of a triple in an arbitrary configuration they have only provided a computationally demanding numerical searching algorithm. Nevertheless, they have implemented techniques to relate the scene and model triple correspondences in an object independent coordinate system. Dhome et al. [25] solved the pose recovery problem using three lines for a wider class of configurations. They have derived an eighth degree polynomial for solving the problem of interpreting a triplet of spatial edges. Dhome's formulation to this problem is quite different from our, in that, the geometric relationships...
between the model and scene triple correspondences are not explicitly defined by the angles between spatial edges. In contrast, we have explicitly related the geometric constraint of the model and scene corresponding triples using the angles between spatial edges. The advantages of using this representation are its simplicity in object modelling and organization.

Furthermore, Dhome [25] did not report how the model-scene correspondences can be established for different classes of configuration. For this problem, we have derived an eighth degree polynomial for interpreting a triple of space lines in general configuration. Our formulation of this problem is novel and more intuitive than existing methods. The polynomial constraint equation is derived using explicitly the space angles between triple edges. The framework presented here is a true shape-from-angle paradigm.

Many researchers attempted to solve the problem of pose determination for a trihedral vertex from its 2D projection. This is a special case of the problem of analysing a general triple of spatial edges. The solution of the above specific problem is fully included in our derivation. Both Kanade [42] and Kanatani [43] tackled this problem by assuming the 3D vertex being viewed under orthographic projection. Kanade [42] solved the problem using an analytical approach. Whereas Kanatani [43] provided a very compact and explicit analytical solution for this problem. However, the applicability of their solutions is limited only to images with weak perspective effects. In other words, their methods are only applicable when the focal length of the camera is large or the object is relatively far away from the camera. This constraint on their methods is not acceptable in a general vision environment where the variation of the range of views is generally very large. For this problem, the results presented here are not restricted to orthographic projection.

Barnard [42] studied the same problem under perspective projection using the Gaussian sphere to represent the geometric constraints and derived an iterative numerical solution. His solution is only applicable to a rectangular corner where the angles between a triple of model edges are orthogonal to each other. Clearly, their object domain is limited. Kanatani [44] and Shakunaga and Kaneko [64] exploited the orientation angles of 3D vertex edges and their projection to solve the pose determination problem which they referred to as shape from angle paradigm. Shakunaga et al. [64] derived a more general approach to the problem of analysing a triple of spatial edges by using the concept of perspective moving coordinate systems. In the case of analysing trihedral vertex, they provided an analytical solution for a trihedral vertex under the assumption that at least two spatial edges are right angle. To tackle this problem, Kanatani [44] developed a technique mapping the crossing point or pencil of the three image lines into the image origin, calling it a canonical position. Using this technique, he also derived an analytical solution for this special type of triple. Notwithstanding these limitations, we find his framework is most elegant and intuitive. Horaud [34] developed a new method for recovering the pose of a scene object using the geometric relationship binding the corresponding model and scene triplets without any restriction on the space angles between edges. Feasible solutions were searched for on a tessellated Gaussian sphere. Although his method is computationally intensive, it is first significant work to integrate the trihedral vertex into a classical
hypothesis and verification paradigm. Later, Horaud et al. [37] provided a quartic equation in closed form, for recovering the pose of a trihedral vertex of general configuration. Roots of such equation can be either solved analytically or using an iterative numerical method. In this respect, we have also derived a quartic equation for interpreting a trihedral vertex with no restriction on angles between the spatial edges. However, the work reported here is the first to provide a compact analytical solution for interpreting a trihedral vertex under perspective projection, in an object-independent coordinate system. This has important implications on model invocation and matching efficiency.

Lowe [50] experimented with groups of 2D non-accidental viewpoint independent cues in his SCERPO system with a view to reduce substantially the number of inconsistent matches that may be considered in the matching process. Having selected subsets of informative features from the image, he proposed an iterative method to estimate the pose of the scene objects by refining the chosen initial transform parameters using a progressively more greater number of hypothesized model-scene correspondences. However, the assignment of the initial pose is a non-intuitive task.

This chapter is organized as follows. In the next section, the new formulation of the problem is described. In Section 6.4, the technique of analysing geometric features imaged perspective at a canonical position is discussed. In Section 6.5, a method for interpreting a triple of space lines is described. In Section 6.6, closed form solutions are derived for some special and common triple configurations. In Section 6.7, methods of computing relative rotation and translation using the recovered edge orientations are discussed. In Section 6.8, the effectiveness of using composite feature to reliably recover the translation and significantly prune down the number of hypotheses generated from model-scene trihedral vertices is discussed. In Section 6.9, an effective constraint equation generation scheme is proposed. In Section 6.10, the stages of a proposed integrated recognition system are described. Section 6.11 contains a description of experiments performed to study the feasibility, effectiveness and robustness of the object recognition approach based on triples of connected spatial edges and trihedral vertices. Finally, in Section 6.12, the main issues of the proposed approach are summarized and conclusions are drawn.

### 6.3 Problem Formulation

Consider three 3D straight edge segments $M_{i=1,2,3}$ the orientation of which is specified by unit directional vector $E_i$ measured with respect to an inherent object model coordinate system $\mathcal{F}_M$ (see Figure 6-2 (a)). Let $\alpha_{ij}(i \neq j)$ be the 3D spatial angle between the unit vectors $E_i$ and $E_j$. Consider the three corresponding scene lines $L_i$ (the three model edge fragments $M_i$ undergoing a rigid motion transformation) of unit directional vectors $S_i$ measured with respect to a camera coordinate system $\mathcal{F}_C$ (see Figure 6-2 (b)). Let $l_i$ be the unit directional vector of the perspective projection of these scene lines $L_{i=1,2,3}$. The two related problems can now be defined as follows:

- **Edge Orientation Problem**: given the 3D space angles $\alpha_{ij}$, the vector $l_i$ of the perspective image line and the intrinsic camera parameters, compute all the geometrically admissible con-
Analysis of a triple of spatial edges

figurations of $S_i$.

- Having computed the feasible sets of solutions of $S_i$, estimate, using the knowledge of model edges $M_i$, the relative rotation $R_{MC}$ and the translation vector $T_{MC}$ to define the relationship between the object model frame $F_M$ and camera frame $F_C$.

It is interesting to note here that the process of estimating the pose of the scene object is broken down into two stages. The above problems are embedded into a single complicated problem in Dhone's formulation [25]. Hence the extensibility of his approach to cope with the problems posed by a large model-base is limited. In contrast, the advantages of our approach can overcome this problem by using generic feature representation and sharing the processing of smaller subgoals.

To solve the first problem, the 3D space angles $\alpha_{ij}$ for $i \neq j$ are the only knowledge required from the model description. To date, we believe the work reported here is the first to formulate the triple interpretation problem explicitly based on the space angles between edges, and yet obtain a compact polynomial solution. The approach described is a true shape-from-angle paradigm. This advances significantly the results of Shakunaga [64] who, using this paradigm, only proposed a numerical searching algorithm for the solution of this problem which is very computationally demanding.

The merit of this representation can be easily realized by the following example. Suppose there are $Q$ model triplets which have the same 3D space angles $\alpha_{ij}$. Given the description of a triple of image lines, the computation load for determining the solution of $S_i$ is independant of the size $Q$ of the triple group. In general computer vision, one will need to deal with a large number of models which will possess features which share the same subsets of space angles. In the proposed approach, the geometric constraint solution is obtained only once for each subset. As a result, the overall efficiency of the hypothesis generation phase of the proposed recognition system will be increased significantly. Furthermore, many implausible candidates generated from the hypothesis phase can be swiftly pruned away using the geometric constraints imposed by $S_i$ computed from different sets of three image lines. The above mentioned hypothesis generation phase and pruning process can be performed without the need to complete pose determination; to involve the quantitative information measured with respect to the inherent object coordinate system. The details and effectiveness of this hypothesis generation scheme will be described in Section 6.9. Any hypothesis generation procedure involving the solution of the pose rotation and translation transforms for every corresponding model-scene feature pair is bound to become impracticable for any moderate size of object model base. Thus breaking down the problem into smaller subgoals is a novel and computationally crucial aspect of our object recognition strategy. First, we will describe the proposed method to solve the edge orientation problem.

6.4 Analysis of scene features in the canonical position

The analysis of 3D geometric features being viewed under perspective projection in general position is very complicated. Hence the geometric insight of the equations derived from these formulations
Analysis of a triple of spatial edges

is in general implicit and non-intuitive. To simplify the analysis of 3D scene features from a single perspective view, Kanatani [44] proposed a technique to move the projection of the target scene feature into an image origin, namely the canonical position where the analysis of the target scene feature can be greatly simplified and made tractable. This canonical configuration is obtained by pointing the optical axis of the camera to the interesting location of the scene feature using a standard camera rotation. For example in Figure 6-1 (a), the image angle of the projection of a spatial 3D edge of unit direction vector $N$ observed under perspective ($\phi_p$) and orthographic ($\phi_o$) are different in general position. However, the distinction between these two disappear ($\phi_p^* = \phi_o^*$) in the canonical position where the optical axis of the camera intersects the point A along the 3D edge. At this canonical position, the directional unit vector ($\vec{N} = R_{CA} N$) of the 3D edge can be simply expressed as $\vec{N} = (\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta)$, where $\theta$ is the space angle between the directional vector $\vec{N}$ and the optical axis of the rotated camera, and $\phi$ is an orientation of the edge under perspective or orthographic projection at the canonical position $F_A$. The standard transformation $R_{CA}$ is a rotation matrix which maps the projection of the interesting point $P_a = P_a(x, y)$ to the origin of the image plane of the camera frame $F_C$. On the other hand, this means the transformation $R_{CA}$ moves the optical axis currently pointing at the interesting point A (at canonical position) to the origin of the image plane of $F_C$. In the rest of the paper, the corresponding image transformation induced by the camera rotation $R$ is denoted by $T$. The details of the derivation of $R$ and $T$ can be found in the interesting work by Kanatani [44]. However, for the sake of completeness, the expressions of these transformations are included in Appendix A.

6.5 Solving the Edge Orientation Problem

In canonical coordinate frame $F_A$

Let us return to the three 3D scene edge fragments $L_{i=1,2,3}$ of unit directional vectors $S_{i=1,2,3}$ and their perspective projection $l_i$ shown in Figure 6-2 (b). Let $P_a$ be the intersection point between the lines $l_1$ and $l_2$ at the camera frame $F_C$. First, a standard transformation $R_{CA}(P_a, f)$ is employed to map the image point $P_a$ at $F_C$ to the origin of the image plane of the canonical frame $F_A$. The optical ray of the canonical frame $F_A$ intersects both the 3D spatial point $(A_1)$ and $(A_2)$ along the scene lines $L_1$ and $L_2$ respectively, and their image point $P_a^*$ (see Figure 6-2(c)).

The unit directional vector $N_i$ of 3D edge fragments $L_i$ at $F_A$ can be computed from $N_{i=1,2} = (\sin \theta_i^A \cos \phi_i^A, \sin \theta_i^A \sin \phi_i^A, \cos \theta_i^A)$, where $\theta_i^A$ is the angle between edge $L_i$ and the optical axis of the canonical frame $F_A$. The image angle $\phi_i^A$ is defined by the orientation of the image line $J_i$ measured with respect to the X-axis of the image plane of $F_A$, where $J_i$ is obtained by applying the image transformation $T_{CA}$ to the image line $l_i$ at $F_C$. The space angle $\alpha_{12}$ between the 3D edge fragments $L_1$ and $L_2$ measured at the canonical frame $F_A$ can be expressed as:

$$N_1 \cdot N_2 = \cos(\alpha_{12})$$
Analysis of a triple of spatial edges

A spatial edge $Z_{ui}$

A camera frame $F_c$

Figure 6-1: Analysis of a spatial edge in general and canonical position

\[
\cos(\phi_1 - \phi_2) \sin(\theta_1^2) \sin(\theta_2^2) + \cos(\theta_1^2) \cos(\theta_2^2) = \cos(\alpha_{12})
\]  

(6.1)

The angle $\alpha_{12}$ can be expressed as $\cos^{-1}(E_1 \cdot E_2)$, where $E_{i=1,2}$ are the unit direction vectors of the model edges $M_{i=1,2}$ corresponding to the scene lines $L_{i=1,2}$, respectively. Once the unit vector $N_i$ of the edge $L_i$ at $F_A$ is determined, the unit directional vector $S_i$ of the corresponding scene edge can easily be recovered by transforming $L_i$ from the canonical frame $F_A$ back to the camera coordinate system $F_C$ using $S_i = R_{CA}(P_{a,f}) N_i$. Next, we will derive another two equations by analysing the three scene lines at the second canonical position.

In canonical coordinate frame $F_B$

Let $P_b^*$ be the intersection of the perspective image lines $J_2$ and $J_3$ in canonical frame $F_A$ (see Figure 6-2 (c)). The intersection point $P_b^*$ at $F_A$ can also be computed by applying the image transformation $T_{CA}$ to the image point $P_b$, which is the intersection of the image lines $l_2$ and $l_3$ at the camera frame $F_C$ (see Figure 6-2 (b)). Having computed the intersection $P_b^*$, a standard transformation $R_{AB}(P_b^*, f)$ is applied to map the image point $P_b^*$ at the frame $F_A$ to the origin $P_b^{**}$ of the image plane of the canonical frame $F_B$. Figure 6-2 (d) shows the configuration of the three lines measured with respect to the canonical frame $F_B$ of which the optical axis intersects the 3D spatial
Analysis of a triple of spatial edges

Figure 6-2: An analysis of a triple of spatial edges in the camera $\mathcal{F}_C$, canonical $\mathcal{F}_A$ and $\mathcal{F}_B$ frame
Analysis of a triple of spatial edges

point \((B_2)\) and \((B_3)\) along the scene lines \(L_2\) and \(L_3\) respectively, and their image plane intersection \(P_b^*\). At the canonical frame \(F_B\), the unit directional vectors \(W_i\) of the scene edges \(L_i\) can be written as follows:

\[
W_i = \mathcal{R}_{AB}^i (P_b^*, f) \cdot \mathbf{N}_i
\]

\[
W_{3} = (\sin \theta_3^2 \cos \phi_3^2, \sin \theta_3^2 \sin \phi_3^2, \cos \theta_3^2)
\]

where \(\theta_3^2\) is the angle between the edge \(L_3\) and the optical ray of \(F_B\) and \(\phi_3^2\) is an orientation of the corresponding image line \(g_3\) at \(F_B\) (see Figure 6-2 (d)). Having expressed the unit vectors \(W_i\), the 3D space angles \(\alpha_{23}\) and \(\alpha_{13}\) can be written as:

\[
W_2 \cdot W_3 = \mathcal{R}_{AB}^2 N_2 \cdot W_3 = \cos(\alpha_{23}) \\
W_1 \cdot W_3 = \mathcal{R}_{AB}^1 N_1 \cdot W_3 = \cos(\alpha_{13})
\]

(6.4)

Since, the 3D space angles are preserved under rotation transformation, Eqs (6.4) can be rewritten as:

\[
N_2 \cdot \mathcal{R}_{AB} W_3 = \cos(\alpha_{23}) \\
N_1 \cdot \mathcal{R}_{AB} W_3 = \cos(\alpha_{13})
\]

(6.5)

In the standard transformation \(\mathcal{R}_{AB} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}\), the elements \(r_{ij}\) are functions of the image point \(P_b^*\) at \(F_A\) and the focal length \(f\) of the camera, which is assumed to be unchange during the process of transformation. The angles \(\alpha_{13}\) and \(\alpha_{23}\) can be expressed as \(\cos^{-1}(E_1 \cdot E_3)\) and \(\cos^{-1}(E_2 \cdot E_3)\), respectively. For the sake of simplicity, we remove the superscripts of the edges and their image line orientations at the canonical frames by rewriting \(\theta_{i=1,2}^A = \theta_{i=1,2}, \theta_3^A = \theta_3, \phi_{i=1,2}^A = \phi_{i=1,2}\) and \(\phi_3^A = \phi_3\). Substituting for \(\mathcal{R}_{AB}, N_{i=1,2}\) at the canonical frame \(F_A\) and Eq. (6.3) into Eq. (6.5) yields,

\[
(K_1(\phi_2) \sin \theta_2 + K_2 \cos \theta_2) \sin \theta_3 + (K_3(\phi_2) \sin \theta_2 + r_{33} \cos \theta_2) \cos \theta_3 = \cos \alpha_{23}
\]

(6.6)

\[
(K_1(\phi_1) \sin \theta_1 + K_2 \cos \theta_1) \sin \theta_3 + (K_3(\phi_1) \sin \theta_1 + r_{33} \cos \theta_1) \cos \theta_3 = \cos \alpha_{13}
\]

(6.7)

where

\[
K_1(\phi_{i=1,2}) = (r_{22} \sin \phi_i + r_{12} \cos \phi_i) \sin \phi_3 + (r_{21} \sin \phi_i + r_{11} \cos \phi_i) \cos \phi_3
\]

\[
K_2 = r_{32} \sin \phi_3 + r_{31} \cos \phi_3
\]

\[
K_3(\phi_{i=1,2}) = r_{23} \sin \phi_i + r_{13} \cos \phi_i
\]

Having derived the system of trigonometric equations (6.1), (6.6) and (6.7), the geometric constraint problem of interpreting a triplet of 3D edge fragments is now formulated as solving for the
Analysis of a triple of spatial edges

three unknowns \( \theta_i, 0 < \theta_i < \pi \) (for \( i = 1, 2, 3 \)).

First, we will derive a polynomial \( P(\theta_1) \) for \( \theta_1 \). Having solved the polynomial \( P(\theta_1) \), the corresponding solutions for \( \theta_{i=2,3} \) can easily be determined. The Constraint Angle Solution (CAS) set \( A^\theta = (\theta_1, \theta_2, \theta_3) \) must satisfy \( 0 < \theta_{i=1,2,3} < \pi \).

To derive a polynomial \( P(\theta_1) \), we will first eliminate \( \theta_3 \) from Eqs. (6.6) and (6.7). Using the resultant equation in two unknowns \( \theta_{i=1,2} \) and Eq. (6.1), a polynomial \( P(\theta_1) \) is derived by eliminating \( \theta_2 \). The details of the method will be described in the next subsection.

6.5.1 Deriving polynomial \( P(\theta_1) \)

To eliminate \( \theta_3 \) from Eqs. (6.6) and (6.7), these two equations can be interpreted as a linear system in the two unknowns \( \cos \theta_3 \) and \( \sin \theta_3 \). The two unknowns can be determined as polynomials of \( \theta_1 \) and \( \theta_2 \),

\[
\begin{align*}
\sin \theta_3 &= F_1(\theta_1, \theta_2) \\
\cos \theta_3 &= F_2(\theta_1, \theta_2)
\end{align*}
\]

(6.8)

(6.9)

where

\[
\begin{align*}
F_{i=1,2}(\theta_1, \theta_2) &= v_{i1} \sin \theta_1 + v_{i2} \cos \theta_1 + v_{i3} \sin \theta_2 + v_{i4} \cos \theta_2 \\
F_3(\theta_1, \theta_2) &= v_{31} \sin \theta_1 \sin \theta_2 + v_{32} \cos \theta_1 \sin \theta_2 + v_{33} \sin \theta_1 \cos \theta_2
\end{align*}
\]

The expressions for \( v_{i=1,2,3} j=1,...,4 \) are given in the Table B.1 in Appendix B. Using the trigonometrical identity, \( \sin^2 \theta_2 + \cos^2 \theta_2 = 1 \), a non-linear trigometric equation in two unknowns \( \theta_{i=1,2} \) can be derived to yield,

\[
F_1^2(\theta_1, \theta_2) + F_2^2(\theta_1, \theta_2) = F_3^2(\theta_1, \theta_2)
\]

(6.10)

After some manipulation, Eq (6.10) can be expressed as follows,

\[
\begin{align*}
\mu_0 \sin^2 \theta_1 \sin^2 \theta_2 + \mu_1 \cos^2 \theta_1 \sin^2 \theta_2 + \mu_2 \cos^2 \theta_1 \sin^2 \theta_2 + \mu_3 \cos \theta_1 \sin \theta_1 \sin^2 \theta_2 \\
+ \mu_4 \sin \theta_2 \cos \theta_2 \sin^2 \theta_1 + \mu_5 \sin \theta_1 \cos \theta_1 \sin \theta_2 \cos \theta_2 + \mu_6 \sin^2 \theta_2 + \mu_7 \sin^2 \theta_1 \\
+ \mu_8 \cos^2 \theta_2 + \mu_9 \cos^2 \theta_1 + \mu_{10} \sin \theta_1 \sin \theta_2 + \mu_{11} \sin \theta_2 \cos \theta_2 + \mu_{12} \cos \theta_1 \sin \theta_2 \\
+ \mu_{13} \cos \theta_2 \sin \theta_1 + \mu_{14} \cos \theta_1 \sin \theta_1 + \mu_{15} \cos \theta_1 \cos \theta_2 = 0
\end{align*}
\]

(6.11)
Analysis of a triple of spatial edges

The expressions for $\mu_{i=1,\ldots,15}$ are given in Table B.2 in Appendix B. The basic terms $\sin \theta_i$ and $\cos \theta_i$ of the trigonometric equation (6.11) can be replaced by the following $t$-formulae,

$$\cos \theta_i = \frac{1 - t_i^2}{1 + t_i^2}; \quad \sin \theta_i = \frac{2t_i}{1 + t_i^2}; \tag{6.12}$$

where $t_i = \tan \frac{\theta_i}{2}$, for $i = 1, 2$. After some manipulation and clearing denominators, we obtain an equation of the form,

$$\delta_4(t_1) \ t_1^2 + \delta_3(t_1) \ t_1^3 + \delta_2(t_1) \ t_2^2 + \delta_1(t_1) \ t_2 + \delta_0(t_1) = 0 \tag{6.13}$$

where $\delta_{i=0,\ldots,4}$ are polynomials of $t_1 = \tan \frac{\theta_1}{2}$ which can be written as,

$$\delta_{q=0,4}(t_1) = \sum_{i=0}^{4} (-1)^{(1-i)} \sigma_{i+1} \ t_1^i \quad \text{for } k = (-1)^{\frac{q}{2}} (4 - i - q)$$

$$\delta_{q=1,3}(t_1) = \sum_{i=0}^{4} (-1)^{(1+i)} \frac{t_1}{4} \sigma_{i+1} \ t_1^i \quad \text{for } k = (-1)^{\frac{q+1}{2}} (2 (3 - q) - i)$$

$$\delta_2(t_1) = \sum_{i=0}^{2} \frac{4 + i (1 - i)}{4} \sigma_{2i} \ ((-t_1)^i + t_1^{4-i})$$

The values of $\sigma_q$ are given in Table B.3 in Appendix B. Next, we will derive another equation containing the two unknowns $t_1$ and $t_2$. As before, the basic terms $\sin \theta_{i=1,2}$ and $\cos \theta_{i=1,2}$ (As mentioned before, $\theta_{i=1,2}$ is relabelled as $\theta_{i=1,2}$) of Eq (6.1) can also be replaced by the $t$-formulae (6.12) to yield the following equation,

$$\rho_2(t_1) \ t_2^2 + \rho_1(t_1) \ t_2 + \rho_0(t_1) = 0 \tag{6.14}$$

where, the $\rho_{i=0,1,2}(t_1)$ are polynomials of $t_1$ and are expressed as,

$$\rho_2(t_1) = (\cos \alpha_{12} + 1) + (\cos \alpha_{12} - 1) \ t_1^2$$

$$\rho_1(t_1) = -4 \cos(\phi_1 - \phi_2) \ t_1$$

$$\rho_0(t_1) = (\cos \alpha_{12} - 1) + (\cos \alpha_{12} + 1) \ t_1^2$$

Having derived the two non-linear equations (6.13) and (6.14) in two unknowns $t_1$ and $t_2$, the polynomial $P(\theta_1)$ can then be determined by eliminating $t_2$ from Eq. (6.14). For the sake of brevity, we drop the argument $t_1$ of $\delta_{i=0,\ldots,4}(t_1)$ and $\rho_{i=0,1,2}(t_1)$ in the subsequent derivation. First considering
Analysis of a triple of spatial edges

Eq. (6.14), we get $t_2 = - \frac{\rho_0 \rho_1 + \rho_0}{\rho_2}$. Substituting this expression for $t_2$ into Eq.(6.13) yields,

$$ t_2 = \frac{\rho_0 (\delta_1 (\rho_1^2 - \rho_0 \rho_2) - \rho_1 \rho_2 \delta_3) + \rho_1^2 (\rho_0 \delta_2 - \rho_2 \delta_0)}{\rho_1 \delta_1 (2 \rho_0 \rho_2 - \rho_1^2) + \rho_2 \delta_3 (\rho_1^2 - \rho_0 \rho_2) + \rho_2^2 (\rho_2 \delta_1 - \rho_1 \delta_2)} $$

(6.15)

Substituting this expression for $t_2$ back into Eq.(6.14) and clearing the denominators, we obtain

$$ \rho_0^3 (\rho_2 \delta_2^3 + \rho_0 \delta_2^2) + \delta_2 \rho_2 (\rho_1 \delta_0 (\rho_0 \rho_2 - \lambda) + \rho_0 (\lambda \delta_1 - \rho_0 \rho_1 \delta_3)) $$

$$ + \delta_1 (\delta_0 (\lambda^2 - 2 \rho_0 \rho_2^2) + \rho_0 (\rho_1 \delta_1 (\rho_0 \rho_2 - \lambda) + \rho_0 (\lambda \delta_2 - \rho_0 \rho_1 \delta_3))) $$

$$ + \rho_0^2 (\delta_2 (\lambda \delta_0 - \rho_0 \rho_1 \delta_1) + (\rho_2 \delta_0^2 + \rho_0 \delta_2^2) + \delta_1 \rho_2 (\rho_0 \delta_1 - \rho_0 \delta_0)) = 0 $$

(6.16)

where $\lambda = \rho_1^2 - 2 \rho_0 \rho_2$. Substituting the polynomials $\delta_i = 0, 1, 4$ and $\rho_i = 0, 1, 2$ into Eq.(6.16), we obtain a polynomial equation of degree 16 in one unknown $t_1$,.

$$ \sum_{i=0}^{16} a_i t_1^i = 0 $$

(6.17)

After extensive analysis, reveals that the polynomial has a very special form :

The coefficient of the even terms of the polynomial are symmetric about the middle one whereas those of the odd terms are anti-symmetric about the middle one ( a sign difference )

The above key observation can be expressed as,

$$ \omega_i = (-1)^i \omega_{16-i} \text{ for } i = (0, \ldots, 7) $$

(6.18)

It is worth noting that the above mentioned form of the polynomial equation is different from the form of reciprocal equation where all the coefficients are symmetric about the middle one. The method of solving ( or reducing the degree of ) reciprocal equation can be found in Barbeau [4]. As far as we could ascertain, there is no classical method for reducing the degree of such a polynomial. To solve this problem, we modify the classical reciprocal substitution method given in by Barbeau [4]. Our modification is simple, but it has not been easy to discover! The modified method will be detailed in the following sections.

First, the polynomial Eq. (6.17) is divided by the variable of the middle term, in this case, $t_8$. The resultant equation can be rearranged to yield the following equation,

$$ \omega_8 + \sum_{i=1}^{8} \omega_{8-i} \left( \frac{1}{t_1^i} + (-t_1)^i \right) = 0 $$

(6.19)

We then use the substitution $Z = (\frac{1}{t_1} - t_1)$. The variable quantity $(\frac{1}{t_1} + (-t_1)^i)$ of equation (6.19)
Analysis of a triple of spatial edges

can then be replaced by $P_i(\mathcal{E})$ and yield the following equation,

$$\omega_8 + \sum_{i=1}^{8} \omega_{8-i} P_i(\mathcal{E}) = 0 \quad (6.20)$$

where $P_i(\mathcal{E})$ are polynomials of $\mathcal{E}$ and can be deduced from,

$$P_0(\mathcal{E}) = 2$$
$$P_1(\mathcal{E}) = \mathcal{E}$$
$$P_{j+1}(\mathcal{E}) = \mathcal{E} P_j(\mathcal{E}) + P_{j-1}(\mathcal{E}) \quad \text{for} \quad j \geq 1.$$

Substituting the polynomials $P_{i=1,\ldots,8}(\mathcal{E})$ into Eq. (6.20), we obtain a polynomial equation of degree 8 in one unknown $\mathcal{E}$,

$$\sum_{i=0}^{8} \psi_i \mathcal{E}^i = 0 \quad (6.21)$$

where,

$$\begin{align*}
\psi_0 &= \omega_8 + 2(\omega_0 + \omega_2 + \omega_4 + \omega_6) \\
\psi_1 &= \omega_7 + 3 \omega_2 + 5 \omega_3 + 7 \omega_4 \\
\psi_2 &= \omega_6 + 4 \omega_4 + 9 \omega_2 + 16 \omega_0 \\
\psi_3 &= \omega_5 + 5 \omega_3 + 14 \omega_1 \\
\psi_4 &= \omega_4 + 6 \omega_2 + 20 \omega_0 \\
\psi_5 &= \omega_3 + 7 \omega_1 \\
\psi_6 &= \omega_2 + 8 \omega_0 \\
\psi_7 &= \omega_1 \\
\psi_8 &= \omega_0
\end{align*}$$

Substituting $t_1 = \tan \frac{\mathcal{E}}{2}$ into the expression $\mathcal{E} = (\frac{1}{t_1} - t_1)$, yields the following result,

$$\mathcal{E} = \frac{2}{\tan \theta_1} \quad (6.22)$$

Substituting this expression for $\mathcal{E}$ into the polynomial equation (6.21) gives the following eighth degree polynomial $P(\theta_1)$ in one unknown, $\tan \theta_1$,

$$P(\theta_1) = \sum_{i=0}^{8} \psi_i \tan^i \theta_1 = 0 \quad (6.23)$$

where $\psi_i = \psi_{8-i} 2^{8-i}$ for $i = 0, \ldots, 8$. For each real root $\theta_1$ obtained from solving the polynomial $P(\theta_1)$, the corresponding solution of $\theta_2$ can be determined from Eq. (6.15) by replacing $\rho_{i=0,1,2} =$
Analysis of a triple of spatial edges

\[ \rho_{i=0,1,2}(\tan \frac{\theta_i}{2}) \text{ and } \delta_{i=0,...,4} = \delta_{i=0,...,4}(\tan \frac{\theta_i}{2}). \]

Another simple alternative is to use Eq. (6.14), the roots of which are

\[ X = \frac{-\rho_1 \pm \sqrt{\rho_1^2 - 4 \rho_2 \rho_0}}{2 \rho_2} \quad (6.24) \]

Given \( 0 < \frac{\theta_i}{2} < \frac{\pi}{2} \), we must choose \( X \) to be positive. Hence

\[ \theta_2 = 2 \tan^{-1} X \quad (6.25) \]

For each corresponding solution for \( \theta_{i=1,2} \), the solution for \( \theta_3 \) can be determined from Eq.(6.8) and Eq.(6.9)

\[ \theta_3 = \tan^{-1} \frac{F_1(\theta_1, \theta_2)}{F_2(\theta_1, \theta_2)} \quad (6.26) \]

In the following, we will call polynomial \( P(\theta_1) \) the Triple Geometric Constraint Equation (TGCE). The coefficients \( \psi_i \) of a TGCE are functions of the following attributes:

- The space angles \( (\alpha_{12}, \alpha_{23}, \alpha_{13}) \) between the 3D edge segments of a model triplet.
- The focal length of the camera.
- The intersections \( P_a \) and \( P_b \) of the corresponding images lines measured w.r.t the camera coordinate system \( \mathcal{F}_C \).
- The orientations \( \phi_{i=1,2} \) and \( \phi_3 \) of the corresponding image lines measured w.r.t the image planes of \( \mathcal{F}_A \) and \( \mathcal{F}_B \), respectively.

The TGCE is established in an object independent coordinate system. At this point, we will introduce some terminology, notation and discuss its meaning. The 3D space angles \( (\alpha_{12}, \alpha_{23}, \alpha_{13}) \) will be called Model Triplet Angle Set (MTAS) and denoted as \( \mathcal{A}^a \). The intersections and orientations of the image lines, denoted as \( \mathcal{A}^l = (P_a, P_b, \phi_{i=1,2,3}) \), will be called the Image Triplet Attribute (ITA). We will assume the setting of the focal length of the camera remains unchanged henceforth. To derive a triplet geometric constraint equation, a MTAS and an ITA are required, we shall call such a pairing as Model-Scene Triplet Correspondence (MSTC).

Dhome [25] derived an eighth degree polynomial in one unknown. However, as mentioned before, they formulated the triple interpretation problem as part of a single non-separable pose determination problem. Hence, it is difficult for their method to cope with more general recognition problems. In the case of interpreting triples of general configurations, Shakunaga [64] only provided a complicated solution involving a numerical searching algorithm. As mentioned before, although our basic approach is inspired by Kanatani [44], the solution derived here can cope with more general and difficult shape recognition problems. Kanatani only provided an analytical solution for a trihedral vertex involving at least two right angles. The extension of his formulation to cope with the more general shape recovery problem is possible but is a non-trivial task. Here, we have demonstrated the formulation and manipulations leading to an analytic solution which can deal with more difficult vision problems.
Having derived the solution for interpreting a general triplet of spatial edges, some special and common spatial configurations of edge triplets will be investigated in the next subsection.

### 6.6 Special and Common Configurations

In this section, three special configurations, namely, generic rectangular bar end, orthogonal and trihedral vertex configurations, are discussed. From our experience, these geometric configurations are often encountered in recognizing industrial components containing straight edges. For these special and common configurations, the *triplet geometric constraint equation* (6.23) simplifies to polynomials of lower degree. Roots of such lower polynomial equations can be solved analytically or using iterative numerical methods. Many of the notable researchers who tackled or derived analytical solutions to subsets of these problems are [25], [44], and [64]. First, we will derive an analytical solution for the case of a *generic rectangular bar end*.

#### 6.6.1 Generic Rectangular Bar End

In this section, the definition and analysis of a *generic rectangular bar end* will be described. For a *generic rectangular bar end*, one of the edge directions of a triplet is orthogonal to the other two. Without loss of generality, the geometric relationship of a *generic rectangular bar end* corresponds to the *model triplet angle set* \( \alpha = (\alpha_{12} = \frac{\pi}{2}, \alpha_{23} = \frac{\pi}{2}, \alpha_{13}) \). By substituting \( \cos \alpha_{23} = 0 \) into the coefficients tabulated in Table B.1 and Table B.2 in Appendix B, the following can be obtained,

\[
\nu_{11} = \nu_{12} = \nu_{21} = \nu_{22} = 0
\]

and

\[
\mu_7 = \mu_9 = \mu_{10} = \mu_{12} = \mu_{13} = \mu_{14} = \mu_{15} = 0
\]

Consequently Eq. (6.11) can be reduced into the following simpler form,

\[
\mu_0 \sin^2 \theta_1 \sin^2 \theta_2 + \mu_1 \cos^2 \theta_1 \sin^2 \theta_2 + \mu_2 \cos^2 \theta_2 \sin^2 \theta_1 + \mu_3 \cos \theta_1 \sin \theta_1 \sin^2 \theta_2 \\
+ \mu_4 \sin \theta_2 \cos \theta_2 \sin^2 \theta_1 + \mu_5 \sin \theta_1 \cos \theta_1 \sin \theta_2 \cos \theta_2 + \mu_6 \sin^2 \theta_2 + \mu_8 \cos^2 \theta_2 \\
+ \mu_{11} \sin \theta_2 \cos \theta_2 = 0
\]

(6.27)

Next, we shall arrange the above equation so that its terms contain only the trigonometric functions \( \tan \theta_1 \) and \( \tan \theta_2 \). One can divide Eq.(6.27) by \( \cos^2 \theta_1 \cos^2 \theta_2 \) and then replace \( \tan \theta_i = \frac{\sin \theta_i}{\cos \theta_i} \). Substituting the trigonometric equivalent \( \frac{1}{\cos^2 \theta_i} = 1 + \tan^2 \theta_i \) into the resultant equation yields the following trigonometric equation in two unknowns \( \theta_i = 1, 2 \).

\[
(\mu_0 + \mu_6) \tan^2 \theta_1 \tan^2 \theta_2 + \mu_3 \tan \theta_1 \tan^2 \theta_2 + (\mu_4 + \mu_{11}) \tan^2 \theta_1 \tan \theta_2 \\
+ \mu_5 \tan \theta_1 \tan \theta_2 + \mu_{11} \tan \theta_2 + (\mu_2 + \mu_8) \tan^2 \theta_1 + (\mu_1 + \mu_6) \tan^2 \theta_2 + \mu_8 = 0
\]

(6.28)
Another trigonometric equation of two variables $\tan \theta_{1,2}$ can be derived. By substituting $\cos \alpha_{12} = 0$ into Eq. (6.1). Simple trigonometrical manipulation yields,

$$\tan \theta_1 \tan \theta_2 = -\frac{1}{\cos(\phi_1 - \phi_2)} = \tau \quad (6.29)$$

Consequently from Eq. (6.29), we obtain $\tan \theta_2 = \frac{\tau}{\tan \theta_1}$. Substituting this expression for $\tan \theta_2$ into Eq. (6.28) yields the following quartic in one unknown $\tan \theta_1$,

$$\sum_{i=0}^{4} H_i \tan^i \theta_1$$

where,

- $H_0 = (\mu_1 + \mu_6) \tau^2$
- $H_1 = (\mu_1 + \mu_3) \tau$
- $H_2 = \mu_8 + \mu_5 \tau + (\mu_0 + \mu_6) \tau^2$
- $H_3 = (\mu_4 + \mu_{11}) \tau$
- $H_4 = \mu_2 + \mu_8$

For each real solution for $\theta_1$, the solution $\theta_2$ can be determined respectively from Eq. (6.29) as,

$$\theta_2 = \tan^{-1} \left( \frac{\tau}{\tan \theta_1} \right) \quad (6.30)$$

and the solution $\theta_3$ can be determined from Eq. (6.8) and (6.9) as,

$$\theta_3 = \tan^{-1} \left( \frac{v_{13} \tan \theta_2 + v_{14}}{v_{23} \tan \theta_2 + v_{24}} \right) \quad (6.31)$$

In summary, we have derived a quartic equation for interpreting the rectangular bar end. In comparison with related work, Shakunaga [64] also provided a quartic equation, but their formulation is no intuitive and is very complicated. Both Kanatani [44] and Horaud [37] derived analytical solutions for a more restricted configuration where three spatial edges must meet at a spatial vertex. Dhoma [25] did not explicitly provide any analytical solution for this configuration. Next, we will derive a closed-form solution for the case of an orthogonal configuration.

### 6.6.2 Orthogonal Configuration

In this section, a triplet of spatial edges of orthogonal configuration will be studied. In this case, all the edges are mutually perpendicular to each other. Thus this case corresponds to a rectangular bar with $\alpha_{13}$ also equal to $\frac{\pi}{2}$. Without loss of generality, we may assume the model triplet angle set $A^\alpha$ to be $A^\alpha = (\alpha_{12} = \frac{\pi}{2}, \alpha_{23} = \frac{\pi}{2}, \alpha_{13} = \frac{\pi}{2})$. By substituting the $\cos \alpha_{12} = \cos \alpha_{23} = 0$ into Eqs. (6.6)
Analysis of a triple of spatial edges

and (6.7), we obtain,

\[ v_{31} \sin \theta_1 \sin \theta_2 + v_{32} \cos \theta_1 \sin \theta_2 + v_{33} \sin \theta_1 \cos \theta_2 = 0 \]

Dividing the above equation by the trigonometric functions \( \cos \theta_1 \cos \theta_2 \) and then replacing \( \tan \theta_{1,2} = \frac{\sin \theta_1}{\cos \theta_2} \), we obtain the following trigometric equation in two unknowns \( \tan \theta_{1,2} = \frac{v_{31}}{v_{32}} \).

\[ v_{31} \tan \theta_1 \tan \theta_2 + v_{33} \tan \theta_1 + v_{32} \tan \theta_2 = 0 \]

Since in this case, \( \cos \alpha_{13} = 0 \), an equation identical to Eq. (6.29) can be derived. From Eq. (6.29), we obtain \( \tan \theta_2 = \frac{\tau}{\tan \theta_1} \). Substituting this trigometric function into Eq. (6.32) and clearing the denominator, we obtain,

\[ v_{33} \tan^2 \theta_1 + v_{31} \tau \tan \theta_1 + v_{32} \tau = 0 \]

The solution for \( \theta_1 \) can be determined using,

\[ \theta_1 = \tan^{-1} \left( -\frac{v_{31} \tau \pm \sqrt{(v_{31}^2 \tau - 4v_{33}v_{32})\tau}}{2v_{33}} \right) \]

For each real solution for \( \theta_1 \), the solutions \( \theta_{i=2,3} \) are given respectively as follows,

\[ \theta_2 = \tan^{-1} \frac{\tau}{\tan \theta_1} \]  

(6.32)

From Eq. (6.7), we obtain

\[ \theta_3 = \tan^{-1} \left( \frac{K_3(\phi_1) \tan \theta_1 + r_33}{K_1(\phi_1) \tan \theta_1 + K_2} \right) \]  

(6.33)

A quadratic equation has been found for describing a triple of orthogonal configuration. Shakanuga [64] has also derived a quadratic equation for this problem. Dhome [25] did not explicitly deal with this special configuration. The analytical solutions derived by both Kanatani [44] and Horwood [37] are only restricted to a rectangular trihedral vertex. Next, we will analyse the configuration of a very interesting feature, namely, a trihedral vertex.

6.6.3 A Trihedral Vertex Configuration

In this section, a trihedral vertex configuration with three space lines emanating from the tip is considered. The notation and geometric configuration of a trihedral vertex are shown in Figure 6-3 (b). In this case, the intersections \( P_a \) and \( P_b \) of the general triplet configuration shown in Figure 6-2 (a) share a common point from which the three image lines radiate. Thus, the two canonical frames \( F_A \) and \( F_B \) in the general triplet formulation are fused together into one canonical position which is coincident with the tip of a trihedral vertex. This means that in the trihedral vertex formulation, the
standard camera transformation $\mathbf{R}_{AB}$ between the two canonical frames $\mathcal{F}_A$ and $\mathcal{F}_B$ is not required, in contrast with the case of general triple. Mathematically, this implies that the standard camera transformation $\mathbf{R}_{AB}$ is replaced with an identity matrix,

$$
\mathbf{R}_{AB} = \begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix}
$$

(6.34)

Substituting the elements $r_{ij}$ of the above matrix into the coefficients tabulated in Table B.1, B.2, and B.3 of Appendix B, we find,

\begin{align*}
\nu_{11} = \nu_{13} = \nu_{22} = \nu_{24} = \nu_{31} = 0 \\
\mu_0 = \mu_3 = \mu_4 = \mu_{11} = \mu_{12} = \mu_{13} = \mu_{14} = 0 \\
\text{and} \\
\sigma_{01} = \sigma_{03} = \sigma_{10} = \sigma_{12} = \sigma_{14} = \sigma_{21} = 0
\end{align*}

Thus, the coefficients $\delta_{i=0,\ldots,4}(t_1)$ of the polynomial (6.13) in $t_1$ can be reduced to a simpler form,

$$
\delta_4(t_1) t_2^4 + \delta_3(t_1) t_2^3 + \delta_2(t_1) t_2^2 + \delta_1(t_1) t_2 + \delta_0(t_1) = 0
$$

(6.35)

where,

$$
\begin{align*}
\delta_{q=0,4}(t_1) &= \sum_{i=0}^{2} \sigma_{0k} t_1^{2i} \\
&\quad \text{for } k = (-1)^{\frac{q}{4}} (4 - 2i - q) \\
\delta_{q=1,3}(t_1) &= \sum_{i=0}^{1} \sigma_{1k} t_1^{1+2i} \\
&\quad \text{for } k = (-1)^{\frac{q-1}{2}} (5 - 2(i + q)) \\
\delta_2(t_1) &= \sum_{i=0}^{2} \sigma_{2k} t_1^{2i} \\
&\quad \text{for } k = 2i (2 - i)
\end{align*}
$$

We can eliminate $t_2$ between Equations (6.14) and (6.35) in a way analogous to the algebraic manipulation performed in the general triplet configuration. Consequently, a polynomial equation of degree 16 with no odd terms in the only unknown $t_1$ is found to be,

$$
\sum_{i=0}^{8} \omega_2 i t_1^i = 0
$$

(6.36)

Analysing the above polynomial, we discover its coefficients are symmetric about the middle term $\omega_5 t_5$. Mathematically, this can be expressed as,

$$
\omega_{2i} = \omega_2 (8-i) \quad \text{for } i = (0, \ldots, 3)
$$

(6.37)
Analysis of a triple of spatial edges

In fact this is a reciprocal polynomial equation, which can be solved by the classical reciprocal substitution method. Dividing Eq. (6.36) by \( t_1^2 \) and together with the symmetry property given in Eq.(6.37), Eq. (6.36) can be rearranged as follows,

\[
\omega_8 + \sum_{i=1}^{4} \omega_{8-2i} \left( \frac{t_i^2}{t^4} + \frac{1}{t^4} \right) = 0
\]  

(6.38)

We now proceed with the classical reciprocal substitution method by writing \( Z = \left( \frac{1}{t^2} + t^2 \right) \). The quantity \( \left( \frac{t_i^2}{t^4} + \frac{1}{t^4} \right) \) of Eq. (6.38) can then be replaced by \( P_i(Z) \) and yields the following quartic equation,

\[
\omega_8 + \sum_{i=1}^{4} \omega_{8-2i} P_i(Z) = 0
\]  

(6.39)

where,

\[
P_0(Z) = 2 \\
P_1(Z) = Z \\
P_{j+1}(Z) = ZP_j(Z) - P_{j-1}(Z) \quad \text{for} \quad j \geq 1.
\]

Substituting the above polynomials \( P_{i=0,..,4}(Z) \) into Eq. (6.39), a quartic equation in unknown \( Z \) is obtained,

\[
\omega_8 Z^4 + \omega_2 Z^3 + (\omega_4 - 4 \omega_0) Z^2 + (\omega_6 - 3 \omega_2) Z + (\omega_8 - 2 \omega_4 + 2 \omega_0) = 0
\]  

(6.40)

Having solved for \( Z \), the solution of \( \theta_1 \) can be obtained as follows,

\[
\theta_1 = \sin^{-1} \left( \frac{4}{Z+2} \right) \quad \text{or} \quad \theta_1 = \pi - \sin^{-1} \left( \frac{4}{Z+2} \right)
\]  

(6.41)

The solution \( \theta_2 \) can be determined from Eq.(6.25). Having computed \( \theta_i=1,2 \), the solution \( \theta_3 \) can be determined from Eq.(6.8) and (6.9) as follows,

\[
\theta_3 = \tan^{-1} \left( \frac{v_{12} \cos \theta_1 + v_{14} \cos \theta_2}{v_{21} \sin \theta_1 + v_{23} \sin \theta_2} \right)
\]  

(6.42)

We have derived a quartic equation for a trihedral vertex of general configuration. In our formulation of this problem, it is very trivial to deal with a special case such as a corner involving at least two right angles or a rectangular corner. We can simply adopt for the model triplet angle set \( \mathcal{A}^a = (\alpha_{12} = \frac{\pi}{2}, \alpha_{23} = \frac{\pi}{2}, \alpha_{13}) \). or model triplet angle set \( \mathcal{A}^a = (\alpha_{12} = \frac{\pi}{2}, \alpha_{23} = \frac{\pi}{2}, \alpha_{13} = \frac{\pi}{2}) \) for the former and latter case respectively. In these situations either quartic or either quadratic equation will be found accordingly. Dhone [25] provided a quartic for a trihedral vertex of general configuration. Horaud et al. [37] revealed a complete analysis for this problem. In their work,
Analysis of a triple of spatial edges

analytical solutions were derived for the trihedral vertex of general and special configurations. Using the shape-from-angle paradigm, Kanatani [44] and Shakunaga [64] also derived analytical solutions for this problem. Shakunaga [64] proposed a solution based on perspective angle transform which is far more complicated than the one proposed by Kanatani. However, Shakunaga's solution can cope with a more general class of shape recovery problems. In this respect, our formulation is more intuitive. Furthermore, the compact solutions derived in this thesis can deal with many general shape recognition problems.

6.7 Pose Estimation

Formally, the problem of pose estimation may be defined as follows: Given a set of N three dimensional vectors with respect to an inherent object model coordinate system, and the 2D perspective projection of the corresponding N vectors of a scene object with respect to the camera frame, estimate the relative rotation $R_{MC}$ and the translation vector $T_{MC}$ to define the spatial relationship between the object model and camera frame. To solve this problem, we first concentrate on solving the relative rotation transform consisting of three degrees of freedom. The translation vector can easily be recovered once the former is determined.

The configuration of the camera model is same as the one employed in Section 5.3. For the sake of brevity, the description of the camera model is included here again. Consider the perspective geometry of a camera model depicted in Fig. 6-3 (b), the image plane is assumed to be in front of the center of projection so as to acquire an upright scene image. The focal length, $f$ is the normal distance from the center of projection to the image plane. Based on the above configuration, the position of the scene vertex $P_s$ can be expressed in a camera frame $F_C$ centered at the origin as $P_s = R_{MC}P_m + T_{MC}$, where $R_{MC}$ is the relative orientation between the model and camera frame and $T_{MC}$ is a translation vector. The corresponding model vertex $P_m$ is measured with respect to an object coordinate system $F_M$. In the next subsections, we will describe methods for computing these parameters.
Analysis of a triple of spatial edges

6.7.1 Relative Rotation

To determine the relative rotation, we decompose the rotation transform into model-to-feature $R_{MF}$ and camera-to-feature $R_{CF}$ transforms.

Model-to-Feature Transform $R_{MF}$

Consider any two non-parallel edges $M_i, i = 1, 2, 3$ of a triplet of unit directional vectors, $E_i, i = 1, 2, 3$, described with respect to an object model coordinate system $F_M$. Without loss of generality, an orthogonal X-, Y- and Z-axes of a basis can be respectively constructed by,

\[ H_1 = E_1 \]
\[ H_2 = E_1 \times E_2 \]
\[ H_3 = \frac{H_1 \times H_2}{||H_1 \times H_2||} \]
Analysis of a triple of spatial edges

Where \( H_i = (h_{ix}, h_{iy}, h_{iz}) \) is a unit vector. The transformation \( R_{MF} \) of vertices \( P_w \) with respect to an object model coordinate system \( F_M \) to vertices \( P_v \) with respect to a feature-based coordinate system \( F_F \) (see Fig. 6-3 (b)) can be expressed as

\[
P_v = R_{MF} P_w \quad \text{where} \quad R_{MF} = \begin{pmatrix} h_{1x} & h_{1y} & h_{1z} \\ h_{2x} & h_{2y} & h_{2z} \\ h_{3x} & h_{3y} & h_{3z} \end{pmatrix} \quad (6.43)
\]

**Camera-to-Feature Transform \( R_{CF} \)**

Likewise, the camera-to-feature \( R_{CF} \) transform can be established in a way similar to the method above. Let us assume the correspondence of the model edges \( M_{i=1,2,3} \) with the scene edges \( L_{i=1,2,3} \). Now consider the unit directional vectors \( S_{i=1,2,3} \) of spatial edge \( L_{i=1,2,3} \) measured with respect to the camera coordinate system \( F_C \).

Orthogonal X-, Y- and Z-axes of a basis can be respectively constructed by the corresponding scene edges \( S_{i=1,2} \),

\[
\begin{align*}
Q_1 &= S_1 \\
Q_2 &= S_1 \times S_2 \\
Q_3 &= \frac{Q_1 \times Q_2}{\|Q_1 \times Q_2\|}
\end{align*}
\]

where \( Q_i = (q_{ix}, q_{iy}, q_{iz}) \) is a unit vector. The transformation \( R_{CF} \) of vertices \( P'_s \) with respect to a camera coordinate system, which is centered at the origin of the object model coordinate system \( F_M \), to vertices \( P_v \) with respect to a feature-based coordinate system can be expressed as

\[
P_v = R_{CF} P'_s \quad \text{where} \quad R_{CF} = \begin{pmatrix} q_{1x} & q_{1y} & q_{1z} \\ q_{2x} & q_{2y} & q_{2z} \\ q_{3x} & q_{3y} & q_{3z} \end{pmatrix} \quad (6.44)
\]

Having determined \( R_{MF} \) and \( R_{CF} \), the rotation transform which maps an object model point \( P_w \) to a scene feature point \( P'_s \) with respect to the camera coordinate system centered at the origin of the model frame \( F_M \), can be written as,

\[
P'_s = R_{MC} P_w \quad (6.45)
\]

where \( R_{MC} = R_{CF}^{-1} \times R_{MF} \). The elements of these transforms can be determined from Eqs. (6.43) and (6.44). In the following, we will call the model-to-camera transform \( R_{MC} \) the relative rotation. Having determined the relative rotation, the translation vector \( T_{MC} \) can be easily recovered. In the next section, we will describe the method of determining the translation vector for the case of a general triplet configuration.
6.7.2 Translation

Let us consider a point on an edge of the object. Let the positional vector of this point measured in the inherent object model coordinate system be denoted by $D_m$. Further, let the positional vector of the same object point in the scene expressed in the camera coordinate system be $D_s$. The translation from the model to scene can be simply expressed as $T_{MC} = D_s + R_{MC}D_m$. Here we will propose a method for determining the translation vector based on our formulation of the problem of interpreting a triple of spatial edges in general configuration. To date, we have not found any similar existing method for solving this problem. This may be due to the formulation of the edge orientation problem.

In the following subsections, the methods and issues of computing the feature vectors $D_s$ and $D_m$ for the case of a triplet of general 3D edges, connected edges and trihedral vertex will be described in detail.

The general case

Without loss of generality, let the spatial points $P_{A2}$ and $P_{A2}$ be the feature vectors measured w.r.t an object and a camera coordinate system, respectively. First, we shall establish in a model coordinate system, the geometric configuration shown in Figure 6-4(a) representing the angular relationships between the triplets of 3D edges $L_{i=1,2}$ and the optical ray $Z_a$ in canonical frame $F_A$ shown in Figure 6-4(b). Likewise, a geometric configuration shown in Figure 6-4(a) will be established in model coordinate system representing the angular relationships between the triplets of 3D edges $L_{i=2,3}$ and the optical ray $Z_b$ in canonical frame $F_B$ shown in Figure 6-4(c). Having derived the geometric configuration, intersection calculations will be performed in the model coordinate system to determine the values of $P_{A2}$ and $P_{A2}$. Next, the details of the method will be described.

Let $<P, V>$ denotes a vector equation of position vector $P$ and direction vector $V$. The geometric relationships between the scene lines $L_1, L_2$ and the optical ray $Z_a$ can be established in a model coordinate system using their angular relationships. Let $M_1 = <P_{m1}, E_1>$, $M_2 = <P_{m2}, E_2>$ and $G_a = <P_a, V_a>$ be the vectors measured in model frame $F_M$ (see Figure 6-4(a)) corresponding to the scene lines $L_1, L_2$ and the optical ray $Z_a$ at $F_A$ (see Figure 6-4(b)). The angles between the model edges $M_1$ and $V_a$, and $M_2$ and $V_a$ must be equal to the angles between the optical ray $Z_a$ and $L_1$, and $Z_a$ and $L_2$, respectively. These angular relationships can be expressed as follows,

$$E_1 \cdot V_a = \cos \theta_1 = N_{1z}; \quad E_2 \cdot V_a = \cos \theta_2 = N_{2z}, \quad (6.46)$$

where $N_{i=1,2}$ are the unit directional vectors of $L_{i=1,2}$ in canonical frame $F_A$. Using the fact that $||V_a|| = 1$ is a unit vector, there are two possible solutions for $V_a$ the above equations. Likewise, the geometric relationships between the scene lines $L_2, L_3$ and the vector $Z_b$ at $F_B$ can be described in the model coordinate system using a similar approach. Let $M_3 = <P_{m3}, E_3>$ and $G_b = <P_b, V_b>$ be the vectors measured in the model frame, corresponding to the scene lines $L_3$ and the ray of vector...
Analysis of a triple of spatial edges

$Z_b$. The angles between the direction vectors $M_2$ and $V_b$, and $M_3$ and $V_b$ must be equal to the angles between the vector ray $Z_b$ and $L_2$, and $Z_b$ and $L_3$ at $F_B$, respectively. These angular relationships can be expressed as follows:

$$E_2 \cdot V_b = \cos \theta^B_2 = W_{2z}; \quad E_3 \cdot V_b = \cos \theta_3 = W_{3z};$$  \hspace{1cm} (6.47)

Using the fact that $\|V_b\| = 1$ is a unit vector, there are two possible solutions for $V_b$ in Eq. (6.47).

Although, there are two possible solutions for solving each $V_a$ and $V_b$. There is only one pair of solutions which will satisfy that their scalar dot product is equal to the cosine of the angle $\beta_{ab}$ between the optical ray $Z_a$ and $Z_b$. This angle constraint can be expressed as:

$$V_a \cdot V_b = \cos \beta_{ab}; \quad \text{where} \quad \beta_{ab} = \tan^{-1} \frac{f}{P^*_b}$$

The solutions $V_a, V_b$ which satisfy the angle constraint will be employed to compute the values $P^m_A$ and $P^*_A$. Let $P_o$ be the intersection between the vectors $G_a$ and $G_b$. The intersections between $G_a$ and $M_1$, $G_a$ and $M_2$, and $G_b$ and $M_3$ can be written as follows:

$$P_o + V_a u_1 = P_{m1} + E_1 u_2$$
$$P_o + V_a u_3 = P_{m2} + E_2 u_4$$
$$P_o + V_b u_5 = P_{m3} + E_3 u_6$$

The above 9 linear equations for the 9 unknowns $P_{ox}, P_{oy}, P_{oz}, u_i = 1, ..., 6$ can be easily solved. Having solved for the unknowns, we can obtain $D_m = P^m_{A2} = P_o + V_a u_3$. The corresponding feature vector in the canonical frame $F_A$ is $D^*_A = (0, 0, \|P_o - P^m_{A2}\|)$. The position vector of the spatial point $P^*_A$ measured with respect to the camera frame can then be recovered by $D_s = R_{CA} D^*_A$. In some cases, the triplet edges under consideration are connected each other. If so, the bulk of intersection calculations performed in the model coordinate system are no longer required. This will increase the computational efficiency of pose determination. Hence, it is worth discussing the method of computing the translation vector in the case of a triplet of connected edges.
Analysis of a triple of spatial edges

(a) Intersection calculation in a model frame

(b) The geometric relationships in canonical frame $\mathcal{F}_A$

(c) The geometric relationships in canonical frame $\mathcal{F}_B$

Figure 6-4: Computation of a translation vector in the general triplet case
The triplet of connected 3D edges case

In the case of connected edges, it is reasonable to assume that the intermediate segment of a triple of connected image lines is the projection of a complete 3D edge. A complete edge is defined as an edge bounded by two 3D spatial vertices. Let the length of the spatial intermediate edge be $L_M$. Now, suppose the triplet of the connected edges is being transformed to one of the two canonical positions where the optical ray intersects one of the vertices of the triplet. Without loss of generality, we assume the triplet is being transformed to the canonical frame $F_A$, where the optical ray intersects the vertex $P_A=A_1=A_2$ and passes through the corresponding image point $P_A^*$ (see Figure 6-5 (b)). Let $P_B^*$ be the perspective image of the other vertex $P_B=B_1=B_2$ of the triplet and $\theta_1$ be the angle between the intermediate edge and the optical axis at the canonical frame $F_A$. Given the configuration of the triplet at $F_A$ shown in Figure 6-5 (b) and using the rule of similar triangles, the distance $D_s$ from the origin along the optical axis to the vertex $P_A$ can simply be expressed as follows,

$$D_s = L_M \left( \frac{f}{\|P_A^* - P_B^*\|} \sin \theta_1 - \cos \theta_1 \right)$$  \hspace{1cm} (6.48)

Where $f$ is the focal length of the camera. Having determined the position vector of $P_A = (0, 0, D_s)$ at the canonical frame $F_A$, its corresponding position vector in the original camera frame $F_C$ can then be easily recovered by $V_s = R_{CA} P_A$. Let the $V_m$ be the position vector of the corresponding vertex measured with respect to a model coordinate system $F_M$. The translation vector from the model to scene can then be computed using,

$$T_{MC} = V_s - R_{MC} V_m$$  \hspace{1cm} (6.49)

A Trihedral Vertex Case

To determine the translation for a trihedral vertex configuration, one of the perspective image line segments radiating from the intersection point must be the projection of a complete edge of a spatial trihedral vertex. It is not easy to hypothesize which of the image line segments corresponds to the complete 3D edge of the image is acquired in an uncontrolled environment. We will defer a detailed discussion of this problem to the next section. Once a hypothesis is made, the translation vector can be determined using methods analogous to the case for a triplet of connected spatial edges. In the formulation, one has to treat the selected plausible image line segment of an image trihedral vertex as an intermediate segment of a triplet in general configuration.

Having determined the complete pose of a triplet of spatial edges imaged perspectively, the results can be employed for predicting the description of the 2D scene features in the verification phase of classical hypothesis-verification paradigm. The details of the matching strategies will be discussed in Section 6.10. First, we shall introduce a feature primitive which can reliably be used for computing the translation vector for the case of the trihedral vertex configuration.
6.8 Composite Vertex-CS feature

To determine the translation from an object model to a scene, one of the three line segments of an image vertex must be the true projection of the edges forming a spatial 3D vertex. In order to select at least one plausible line segment, we introduce a primitive, namely vertex-CS feature, by combining a vertex and a V-junction which share a common line segment as shown in Fig. 6.5(b). This feature description can reliably be extracted from image data. The common line ab of the vertex-CS feature is taken as the true projected 3D edge AB as the V-junctions a and b are most simply interpreted as the projections of the two spatial vertices A and B respectively. The geometric constraint imposed by the vertex-CS feature can be employed to discard the inconsistent hypotheses generated from the model-scene vertex pairs. The reader may point out that the vertex-CS feature can be decomposed into a trihedral vertex and a connected triplet. Then one can ask why don’t we use the connected triplet alone to compute the translation vector instead of both. The reasons can be explained as follows,

- Some of the existing systems were solely based on trihedral vertices as seed or key feature primitives for generating feasible hypotheses. To a certain extent, these systems encounter the same problem in computing the translation vector. To improve the robustness and reliability of these systems, one can easily integrate the formulation derived here into other vision systems.

- A polynomial of eighth degree is derived for a general edge triplet which is higher than in the case of a trihedral vertex configuration where a quartic is obtained. In the former case, roots of the polynomial are solved by using iterative numerical method, whereas the latter can be solved analytically.

- The number of trihedral vertex-CS features extracted from scene images is generally smaller than the number connected edge triplets. A small number of “good quality” image vertex-CS features suffice to accomplish the pose determination and recognition task. This assertion has been verified in the our work [75], [76].

Having discussed the advantages of using the trihedral vertex-CS feature against the connected edge triplet as a key feature for recognition, one may think wonder why we need to use the connected edge triplet at all. As mentioned before, the number of extracted vertex-CS features is generally small. Intuitively, such vertex-CS features are not detected systems which are solely based on this feature will completely fail. In general the number plausible image edge triplets edges extracted from the scene is quite large. Hence the success rate of recognition systems based on connected edge triplets is very high. But on the other hand, the computation and processing time are higher than those systems solely based on vertex-CS features. Hence, we see these two primitive features as complementary rather than opposing each other.

Now, we shall derive the geometric constraints imposed by the composite vertex-CS feature. The edge orientation $\vec{N}_1, \vec{N}_2$ and $\vec{N}_3$ of the vertex at the canonical position a can be determined by solving
the equations described in Section 6.5. All the edge orientations $N_i^*$ are represented as unit vectors. To analyse the V-junction $Q$ of the composite feature, the camera is transformed to a new canonical position by pointing the optical axis to the vertex $Q$. In other words, transforming the V-junction $q$ to the origin of the image plane. Let $R_{AQ}$ be the standard transform which maps the image point $q$ to $a$. The edge orientations $N_1^*, N_2^*$ and $N_3^*$ under the new canonical position $Q$ can then be written as $(R_{AQ} N_1)$, $(R_{AQ} N_2)$ and $(R_{AQ} N_2)$ respectively, where $t$ denotes a transpose. The angle $\alpha_{14}$ of the vertex $Q$ can be expressed as a dot product $N_1^* \cdot N_4^* = \cos(180 - \alpha_{14})$. After some manipulation, the angle $\theta_i^*$ between the edge $N_i^*$ and the optical axis can be expressed as:

$$
\theta_i^* = \cos^{-1} \left( \frac{\cos(180 - \alpha_{14})}{\sqrt{(n_{1x}^* \cos \phi_4^* + n_{1y}^* \sin \phi_4^*)^2 + (n_{1z}^*)^2}} \right) + \tan^{-1} \left( \frac{n_{1x}^* \cos \phi_4^* + n_{1y}^* \sin \phi_4^*}{n_{1z}^*} \right) \tag{6.50}
$$

The edge angle $\theta_i^*$ can be determined by substituting the projected edge angle $\phi_4^*$ and the true 3D angle of the vertex $Q$ into the Eq. (6.50). The solutions can then be substituted into the equation,

$$
\beta = \cos^{-1} \left( \frac{\langle N_1^* \times N_2^* \rangle}{|N_1^* \times N_2^*|} \cdot N_4^* \right) \tag{6.51}
$$

where $\beta$ is the angle between the normal $N_1^* \times N_2^*$ and the edge $N_4^*$. The measured angle $\beta$ should correspond to the pre-computed angle $\beta$ of the hypothesized model. The model-scene vertex pair hypotheses which agree with angle $\beta$ will be considered in the verification process. It is worth noting that the geometric constraint imposed by the composite feature does not require quantitative information about the edge length. In the next section, we will discuss the matching strategies used for recognising polyhedral objects using a single perspective image.

### 6.9 An Effective Hypothesis Generation Scheme

In this section, the effectiveness of the proposed method for deriving constraint equations from model and image triple correspondences are discussed. The scheme is based on sharing common space angles derived from the triples.

The hypothesis generation is explicitly broken down into the problems of the geometric constraint equation generation, the determination of the relative rotation and computing the translation vector. It is pertinent that the necessary notation and its meaning in the context of hypothesis generation is first introduced:

- **Generic Triplet Group** $M_i$ : is a group of model triplet features $M_i = \{m_{i1}, m_{i2}, \ldots, m_{in}\}$ which have the same *model triplet angle set* $A_i^T$. The vector equations of the model triplet feature measured with respect to its inherent object model coordinate system is denoted by $m_{ij}$

- **$M$** : is a set of **Generic Triplet Groups** $M = \{M_1, M_2, \ldots, M_n\}$
Analysis of a triple of spatial edges

- $F$ : is a set of image triplet features $F = \{F_1, F_2, \ldots, F_d\}$ extracted from a given scene image. Each of these image triplet features $F_j$ is described by *image triplet attributes* $A_j$.

- $MTAS_{ij}$ : a triplet geometric constraint equation derived from a given pair of *model triplet angle set* $A^i_f$ and *image triplet attributes* $A_j$.

- $H^0_{ij}$ : a set of *constraint angle sets*, i.e. $H^0_{ij} = \{A^0_{i1}, A^0_{i2}, \ldots, A^0_{ij}\}$ computed by solving $MTAS_{ij}$.

- $H^N_{ij}$ : a set of *constraint edge orientation sets*, i.e. $H^N_{ij} = \{A^N_{i1}, A^N_{i2}, \ldots, A^N_{ij}\}$ computed from the corresponding $H^0_{ij}$ and $A_j$.

- $n(S)$ : denotes the number of elements in the set $S$.

![Diagram](a) A triplet of connected edges  
(b) A composite trihedral vertex-CS feature

Figure 6-5: see text
The algorithm of the effective hypothesis generation scheme is given as follows.

**Effective Hypothesis Generation Scheme**

for \( i = 1, \ldots, n(M) \)

begin

for \( j = 1, \ldots, n(F) \)

begin

P1 : Derive \( MTAS_j \) from \( A_j^p \) and \( A_j^f \)

P2 : Solve for \( H_j^\theta \) from \( MTAS_j \)

for \( k = 1, \ldots, n(H_j^\theta) \)

begin

P3 : Determine \( A_{ijk}^\eta \) from \( A_{ijk}^\theta \) and \( A_j^f \)

P4 : Compute \( R_C F_{ijk} \) from \( A_{ijk}^\eta \)

for \( l = 1, \ldots, n(M_i) \)

begin

P5 : Compute \( R_{M_i} F_{ijk} = R_{C_i F_{ijk}} \ R_{M_i F} \)

end

end

end

It is important to note that the bulk of heavy computations from \( P_1 \) to \( P_4 \) only require to perform \( K \) times for each given generic triplet model group \( M_i \) and a triplet image scene feature, \( A_j^f \). The value of \( K \) does not depend on the number of triplet model features \( m_{ij} \) stored in the triplet group \( M_i \).

In the case of connected triplet edges \( K = 2 \), since either edge (wing) connected to the middle edge of the model triplet can be assigned to the corresponding image triple side edge. However, in some case, these two correspondences yield the same model triplet angle set \( A_j^\eta \). If so, two identical triplet geometric constraint equations are derived for each correspondence. This means the processing steps \( P_1 \) and \( P_2 \) are performed once for each correspondence. A typical example is a coplanar symmetric bar end \( A_j^\eta = (\alpha, 0, \alpha) \)

In the case of trihedral vertices \( K = 3 \), since either one of the 3 edges emanating from a tip of trihedral vertex can be assigned to the corresponding image line radiating from a crossing point. Likewise, in some case, these 3 correspondences yield the identical model triplet angle set \( A_j^\eta \). This means processing steps \( P_1 \) and \( P_2 \) are required to perform once for each correspondence. A typical example is a rectangular corner \( A_j^\eta = (\frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}) \)

Next, we shall perform a complexity analysis of the above scheme. The number of computation
Analysis of a triple of spatial edges

The number of computation steps required for \( P_1 \) and \( P_2 \) in the worst case is,

\[
C(P_1, P_2) = K \cdot n(M) \cdot n(F)
\]  \hspace{1cm} (6.52)

The number of computation steps required for \( P_3 \) and \( P_4 \) in the worst case is,

\[
C(P_3, P_4) = C(P_1, P_2) \sum_i \sum_j n(H_{ij}^\theta)
\]  \hspace{1cm} (6.53)

Both \( C(P_1, P_2) \) and \( C(P_3, P_4) \) do not depend on the number of model triplet features \( m_{ij} \). The number of computation steps required for \( P_5 \) in the worst case is,

\[
C(P_5) = C(P_1, P_2) \sum_i \sum_j n(H_{ij}^\theta) \cdot n(M_i)
\]  \hspace{1cm} (6.54)

From the above analysis, it should be clear that \( C(P_5) \) is the only computation dependant on the size of the triplet model feature \( m_{ij} \). Furthermore, the computation of \( P_5 \) only involves simple addition and multiplication. It is worth noting that when no real solutions \( n(H_{ij}^\theta) = 0 \) are found in solving the roots of a triplet geometric constraint equation, this simply means the geometric relationships between a model triplet and image triplet are not geometrically admissible. Hence, if such a situation is found, the computation steps \( P_{3,4,5} \) are not required.

It is clear that the triplet model features measured w.r.t different model coordinate systems can share the same \textit{generic triplet model group}, as long as these triplets can be described by the same \textit{model triplet angle set} \( A^\theta \). In general, the model base of a polyhedral recognition system will contain many triplet features which share the same \textit{model triplet angle set}. This will certainly reduce the computational load significantly compared to systems which do not break down the hypothesis generation into smaller modules. Our hypothesis generation scheme can be decomposed into small subproblems mainly because of the choice and representation of the key features and the hierarchical organization of the model features. In the next section, we will describe the matching strategies used for recognising polyhedral objects from a single perspective image.

### 6.10 Matching Strategies

In the previous sections, we have discussed and formulated the problems such as derivation of triplet geometric constraint equation and pose determination. We then proposed an effective hypothesis generation scheme and analysed its complexity. These modules are now integrated into our system to accomplish the task of polyhedral object recognition. In this section, we will describe four distinct phases in the proposed recognition system. These distinct phases are model generation, image feature extraction, hypothesis generation and verification. First, the model generation will be presented.
6.10.1 Model Generation

This phase is to extract and organize model edges into triplets for matching against image triplets. Triplets of connected edges are generated from CAD-based models.

In the subsequence of the text, a connected edge triple is simply written as triple. Triplet edges which have the same triplet model angle set are assigned into the same generic model triplet group $M_i$. Having generated all the generic model triplet groups, each of these groups is labelled as one of the following type,

- General triplet.
- Coplanar C-shaped triplet.
- Coplanar S-shaped triplet.
- Symmetric triplet
- Trihedral vertex.

The edges of the model triplets $m_{ij}$ are described by vector equations measured with respect to their inherent model coordinate system. Having generated all the model triplets $m_{ij}$, the model to triplet transforms $R_{MF_{ij}}$ are computed. The method of extracting image connected triplets and trihedral vertex features will be described in the next section.

6.10.2 Feature Extraction

In this section, we briefly present a simple algorithm for grouping 2D line segments into triplets of connected image lines and triplets of three image lines meeting at a crossing point, that correspond to plausible physical 3D structures of polyhedrons. The algorithm starts by identifying junctions created by two line segments whose end points are mutually proximal to within a junction threshold, measured along the line of intersection.

Extraction of triplets

Having extracted the V-junctions from a scene, triples are formed by combining pairs of junctions which share a common line, called an intermediate segment. Each of the extracted image triplets is associated to one of the type attributes, namely, C-triples or S-triple. C-triples must have two end points located on the same half plane with respect to the intermediate segment, whereas the end points of a S-triplets are located on different half planes.

Extraction of trihedral vertex

Three image lines will be identified as a trihedral vertex if they satisfy some predefined criteria such as junction region size, the length of the radiating line segments and the angles between them.
Analysis of a triple of spatial edges

In our current implementation, the average of the intersections computed for the 3 possible image line pairs is employed to define a crossing or common point. In order to control the combinatorial explosion associated with unconstrained association of model-scene vertex pair assignments, high quality vertices with small region size, relatively long segments and reasonable angle sizes between them are extracted from the given scene first.

6.10.3 Hypothesis Generation

Given an image triple and a generic triple group \( M_i \), the following rules must be considered,

- A Coplanar C-shaped model triplet will not be matched to any image S-triple
- A Coplanar S-shaped model triplet will not be matched to any image C-triplet
- For an extracted image triplet, there are generally two possible correspondences for matching against a model triplet of general configuration.
- Clearly that a trihedral vertex will not be matched against either a C-triplet or S-triplet.

Having considered all the image and model feature correspondences with the above rules, a list of valid correspondences will be submitted to an effective hypothesis generation scheme for deriving feasible hypotheses. The details of the process will be described in the next section.

6.10.4 Verification

There are two phases in the verification process. Firstly, simple logical rules are exploited to reject infeasible hypotheses. The role of second phase is to perform a detailed check of the correspondence between remaining hypothesized object models and image data, confirming features present and accounting for features which are not observed. In this process, infeasible hypotheses are pruned away and the most plausible candidates are selected as an instance of the target object. The procedures exploited in the verification process can be described as follows.

Pruning infeasible hypotheses

Hypotheses can be further pruned away by using very simple and logical rules suggested by Lowe [50] and implemented in Dhume’s work [25]. These intuitive rules can be summarized as follows.

- As the target scene object is being viewed in front of the camera, all the hypothesized object’s pose of which at least a vertex of coordinate along the Z-axis is negative can be rejected from further verification.
- The hypothesized objects containing at least one hidden edge which is interpreted as an image triplet edge extracted from the scene are removed from the candidate list.
Analysis of a triple of spatial edges

- The predicted projection of the corresponding triple edges must be overlapping with the image triple line segments. A hypothesized object generated by triple edges which do not satisfy this condition can be rejected from the subsequent process (Dhome [25]).

Using the above logical rules, the number of hypotheses is generally reduced significantly. Next, the 2D predictions of the residual hypotheses are then compared with the 2D geometric primitives extracted from the scene image.

**Identification of the most plausible hypotheses**

The 2D description of each hypothesized object is generated by backprojecting the model using the transformation computed in the hypothesis generation module. The confidence measure for each plausible hypothesis is determined using the method described in Section 5.4.3 of Chapter 5. Having computed the confidence measure of each hypothesis, a confidence measure plot of all hypotheses is generated.

The main assumption of the following proposed method is that a global peak in the confidence measure space will be generated by the target object. This is quite reasonable as long as one can ensure the target object is not severely occluded in a given scene or the segmentation of the image is good. Having generated a confidence measure plot, the hypothesis \( HYP0 \) which has maximum confidence measure \( P_8 \) is identified by searching through the entire list of plausible hypotheses. If the maximum confidence measure \( P_8 \) of \( HYP0 \) is quite significant (say above 60%), this is indicative of the presence of an instance of the target object. All hypotheses associated with a confidence measure greater than a cutoff threshold \( Q_f \times P_8 \) are then selected. The relative rotation and translation errors between the selected hypotheses and \( HYP0 \) are computed. If the error of the hypothesized candidate is within an allowable tolerance, the hypothesis is then stored in the plausible hypothesis list \( H_p \). The pose of the scene object is then computed by averaging the rotation and translation of \( HYP0 \) and the candidates stored in \( H_p \). If the confidence measure \( P_8 \) of \( HYP0 \) is only marginal (say 50%), this might indicate partial occlusion. In this regard, we will still accept the hypothesis if the number of plausible candidates stored in the list \( H_p \) is exceeds \( Q \times total \ number \ of \ model \ triples \ and \ trihedral \ vertex \). This is somewhat like a Hough-clustering approach.
6.11 Experimental Results

The model and camera frames employed in all the experiments are designated as right-handed coordinate system. The same intrinsic camera parameters are used in all the subsequent experiments. All the dimensions of the object models are in (mm).

The first experiment involved of the polyhedral object model shown in Figure 6-6. The vertices of the object model are measured with respect to the right-handed model coordinate system are tabulated in Figure 6-6. A 2D perspective test image is shown in Figure 6-10(a). This was generated synthetically by giving the model an known, arbitrary rotation and translation and then perspectively projecting it onto an image plane of a simulated camera of x-resolution = 0.030mm/pixel and y-resolution = 0.028mm/pixel with the focal length = 12.5mm.

The image was processed by the Canny edge detector and the straight lines were identified using a Hough-based line algorithm [60]. The end points of the selected image lines are tabulated in Figure 6-7 (A). All these readings are measured in pixel values with respect to the center of the image plane. The X- and Y- axis of the image plane point from left to right and top to bottom respectively.

In this example, we simply chose the correct corresponding model edges for matching with the extracted scene lines. Three different combinations of three straight lines were chosen in this test. The model and scene triplet correspondence MSTC are shown in Figure 6-7 (B). For example, in MSTC (1), the image line (1) is matched with the edge between vertices (0, 1). MSTC (1), (2) and (3) correspond to a general triple, an orthogonal and a trihedral vertex configuration, respectively.

Using the method described in Section 6.6, the constraint polynomial equations for MSTC (1), (2) and (3) were derived from equations (6.23), (6.32) and (6.40), respectively. The coefficients of the polynomials arranged in an ascending order are given in Figure 6-8.

Having solved the constraint equations, the computed constraint angle set $A^\theta = (\theta_1, \theta_2, \theta_3)$ at the canonical positions are given in Figure 6-9 (A). For each computed $A^\theta$, the corresponding estimated rotation $R(\Phi_x, \Phi_y, \Phi_z)$, and translation $T(t_x, t_y, t_z)$ are given in Figure 6-9 (B), where $\Phi_x, \Phi_y$ and $\Phi_z$ are the angles of rotation about the X-, Y- and Z-axis respectively. The rotation transformation $R(\Phi_x, \Phi_y, \Phi_z)$ can be written as follows.

$$
\begin{bmatrix}
\cos \Phi_z \cos \Phi_y & \cos \Phi_z \sin \Phi_y - \sin \Phi_z \cos \Phi_x & \cos \Phi_z \sin \Phi_y + \sin \Phi_z \sin \Phi_x \\
- \sin \Phi_y & \cos \Phi_y \sin \Phi_x & \cos \Phi_y \cos \Phi_x \\
\sin \Phi_z \sin \Phi_y & \sin \Phi_z \cos \Phi_y - \cos \Phi_z \sin \Phi_x & \sin \Phi_z \cos \Phi_y + \cos \Phi_z \cos \Phi_x
\end{bmatrix}
$$

In the case of trihedral vertex-CS feature, the intersection of image lines (3) and (8) was selected as a V-junction and was found to be at coordinate (45, 41) in pixel values measured w.r.t the center of the image plane. In this example, there were 9 feasible hypotheses generated from the 3 MSTC. Having transformed the model using the estimated pose, the confidence measure of each hypothesis was computed using the method described in Section 6.10.4.
Analysis of a triple of spatial edges

Representative incorrect hypotheses of MSTC (1), (2) and (3) are shown in Figure 6-10 (c), (d) and (e) respectively. Their corresponding pose estimates in Figure 6-9 (B) is labelled with (φ). The highest confidence measure for each MSTC is labelled with (+). To compute the relative pose errors of the best hypotheses, the following definitions are presented. The relative translation error $\Delta T$ can be defined as,

$$\Delta T = \frac{||T_e - T_g||}{||T_e||}$$ (6.55)

where $T_e$ is the estimated translation vector and $T_g$ is the known ground truth translation vector. The relative rotational error is defined as the root sum squared of the differences between the estimated rotational angles $\Phi^e$ and the ground truth values $\Phi^g$ and is expressed as follows,

$$\Delta \Phi = \sqrt{(\Phi^e_x - \Phi^g_x)^2 + (\Phi^e_y - \Phi^g_y)^2 + (\Phi^e_z - \Phi^g_z)^2}$$ (6.56)

The ground truth of the rotation and translation for the rendered object is $\Phi^g = (140.0, 60.0, 120.0)$ and $T_g = (-20.0, -30.0, 500.0)$, respectively. The relative translation and rotation errors of the best hypotheses were computed using Eqs. (6.55) and (6.56) respectively. The rotational errors of the best hypotheses for each assignment MSTC (1), (2) and (3) were 0.86°, 1.13° and 1.39° respectively. The corresponding relative translation errors were 1.02%, 0.49% and 0.05% respectively. Figure 6-10(f) shows the model superimposed onto the scene image using the transformation determined by averaging the estimated translations and rotations of the best hypotheses.

In this experiment, the target object was successfully identified from the rendered scene image. The estimated poses of the object from the 3 MSTC were located very accurately. This can be easily verified by the small relative pose errors and the closeness of match shown in Figure 6-10 (f).

In the next experiment, we examine the performance of the proposed method with noise added to the perspective projection of synthetically generated triplets of straight lines.

![Figure 6-6: The vertices of a model](image-url)
Analysis of a triple of spatial edges

<table>
<thead>
<tr>
<th>(A)</th>
<th>(B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extracted image lines</td>
<td>MSTC</td>
</tr>
<tr>
<td><strong>Line</strong></td>
<td><strong>End points of image line</strong></td>
</tr>
<tr>
<td>1</td>
<td>(-19, -26) - (-38, 1)</td>
</tr>
<tr>
<td>2</td>
<td>(90, -6) - (39, -44)</td>
</tr>
<tr>
<td>3</td>
<td>(-13, 55) - (43, 41)</td>
</tr>
<tr>
<td>4</td>
<td>(23, -19) - (33, -40)</td>
</tr>
<tr>
<td>5</td>
<td>(19, -15) - (-35, 3)</td>
</tr>
<tr>
<td>6</td>
<td>(-40, 5) - (-18, 53)</td>
</tr>
<tr>
<td>7</td>
<td>(87, -4) - (21, -15)</td>
</tr>
<tr>
<td>8</td>
<td>(91, -1) - (45, 41)</td>
</tr>
</tbody>
</table>

Figure 6-7: Extracted Image lines and their corresponding model edges

<table>
<thead>
<tr>
<th>MSTC</th>
<th>Constraint polynomial equation</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>78.4641, -36.2441, -742.5840, -444.5753, -122.2770, 480.6592, 1244.1583, 546.0589, 554.5015</td>
<td>tan $\theta_1$</td>
</tr>
<tr>
<td>2</td>
<td>-0.5938, -0.1562, 0.2177</td>
<td>tan $\theta_1$</td>
</tr>
<tr>
<td>3</td>
<td>0.1121, -0.8811, 2.6965, -10.2114, 20.662</td>
<td>$2 \left( \frac{\tan \theta_2}{\sin \theta_1} - 1 \right)$</td>
</tr>
</tbody>
</table>

Figure 6-8: Geometric Constraint Equations

<table>
<thead>
<tr>
<th>MSTC</th>
<th>(A)</th>
<th>(B)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$(\theta_1, \theta_2, \theta_3)$</td>
<td>$\Phi_e = (\Phi_x, \Phi_y, \Phi_z)$</td>
</tr>
<tr>
<td>1</td>
<td>(38.80, 171.44, 47.82)</td>
<td>(32.11, 301.55, 133.70)^o</td>
</tr>
<tr>
<td></td>
<td>(158.32, 33.39, 96.36)</td>
<td>(139.41, 60.57, 119.74)^+</td>
</tr>
<tr>
<td></td>
<td>(15.75, 143.52, 65.33)</td>
<td>(67.68, 279.22, 153.15)</td>
</tr>
<tr>
<td>2</td>
<td>(143.09, 117.66, 114.04)</td>
<td>(42.47, 303.81, 114.48)^o</td>
</tr>
<tr>
<td></td>
<td>(26.019, 71.195, 73.836)</td>
<td>(140.80, 60.79, 120.09)^+</td>
</tr>
<tr>
<td></td>
<td>(47.58, 91.45, 70.70)</td>
<td>(139.08, 60.09, 118.96)^+</td>
</tr>
<tr>
<td></td>
<td>(32.42, 88.55, 109.30)</td>
<td>(148.71, 296.84, 70.04)</td>
</tr>
<tr>
<td></td>
<td>(64.28, 16.30, 62.78)</td>
<td>(26.53, 184.49, 220.49)</td>
</tr>
<tr>
<td></td>
<td>(115.72, 163.70, 117.22)</td>
<td>(44.78, 22.76, 44.86)^o</td>
</tr>
</tbody>
</table>

Figure 6-9: The constraint angle solutions and estimated object's poses
Analysis of a triple of spatial edges

(a) Synthetic test image   (b) A line drawing

c) Hypothesis using general three lines   (d) An Orthogonal Configuration

(e) Hypothesis using a trihedral vertex   (f) Final pose estimation

Figure 6-10: The experimental results from synthetic test image
Synthetically generated triplets of straight lines were used in this experiment to examine the sensitivity of the proposed methods to image noise. The perspective test images were obtained by projecting triplets of randomly generated straight lines onto an image plane. The parameters of the simulated camera are the same as in the previous experiment.

For a trial, a triplet of 3D edges is randomly generated within a cube of size 200\(mm\) placed at the center of projection or view point of a camera. A random number generator with a uniform distribution was used to create triplets of 3D edges of different length from the range 100 \(\pm 10\) \(mm\) of different position (within the bounding cube) and of direction vector.

The bounding cube is allowed to slide along the optical axis of the simulated camera, and its depth is measured from the center of the cube to the view point of the camera. The depths are designated as the horizontal axis in the positional and rotational error plots. At each of these depths, over 10,000 triplets of 3D edges were randomly generated in the bounding cube. The perspective image coordinates of the 3D edges were obtained by projecting the randomly generated triplets onto the camera image plane. The perturbation was performed by adding independent uniformly distributed noise with zero mean to both the x and y coordinate of each quantized pixel of the perspective image lines. For each set of the perturbed pixel values of a triplet of image lines, the rotation and translation (in this case the depth of the cube) of the triplet of 3D edges were computed. If multiple solutions exist, the one which is closest to the ground truth was chosen to be plausible candidate.

Having estimated the poses of the simulated scene triplets, the relative rotation and translation errors were computed using Eqs. 6.55 and 6.56, respectively. The simulation results were determined by averaging the computed rotation and translation errors over the 10000 randomly generated triplets of 3D edges within the bounding cube being located at each depth. Figure 6-11 (a) and (b) shows the average rotation and translation error plots computed at different depths of the cube ranging from 200 \(mm\) to 1000 \(mm\) in steps of 50 \(mm\).

Figure 6-11 (a) shows that the relative rotation errors were quite sensitive to the depth of the bounding box measured w.r.t to the origin of the simulated camera. At the noise level of 2 pixels or less, the relative error are less than \(10^5\). In the case where the depth of the bounding box is at the distance 400 \(mm\) or less, the rotation error is always less than \(10^5\).

The computed relative translation error or positional error are shown in Figure 6-11 (b). The results show that the average translation error is almost independent of the depth of the bounding box when the noise level is 2 pixels or less. At this noise level, the translation errors are always less than 6%.

The results show that the proposed method is feasible for localizing an object from a wide range of view points using a single perspective image, as long as the image feature coordinate pixel value can be kept within 2 pixels. For a short object range, (say less than 400 \(mm\)) the accuracy of the pose estimates method can tolerate a larger image coordinate perturbation. Next, a real image will be employed to test the sensitivity and robustness of the pose determination and verification scheme.
Analysis of a triple of spatial edges

(a) The relative rotation errors

(b) The relative translation errors

Figure 6-11: Perturbation: \(\triangle: 0.5\) \(+: 1.0\) \(\triangleq: 1.5\) \(\times: 2.0\) \(\Diamond: 2.5\) \(*: 3.0\)
Analysis of a triple of spatial edges

In this experiment, real images were employed to test the robustness and feasibility of the proposed methods for object identification and pose estimation.

In the test scene, a pyramid model was placed on top of the hexagonal polyhedral model, so that a part of the hexagonal model was occluded. The test images were taken with a standard CCD camera. Figure 6-16(a) and (b) shows the gray-level intensity image and straight lines extracted by Hough process [60], respectively. The end points of the selected scene lines are measured in rounded integer pixel values with respect to the center of the image plane and are given in Figure 6-13 (A).

In this example, 5 model and scene triple correspondences (MTAS) were selected and are given in Figure 6-13 (B). The triplet of image lines (1, 2, 3) and (3, 7, 8) corresponds to connected triples of the hexagonal polyhedral model of coplanar and non-coplanar configurations, respectively. The image triple (4, 5, 6) corresponds to a connected triple of the hexagonal polyhedral model of generic rectangular bar end configuration. In the case of the pyramid model, two triplets of extracted image lines (9, 10, 11) and (9, 10, 12), corresponding to a connected triple and a trihedral vertex of general configuration were employed in the pose determination.

It is worth noting that the image lines (6) and (7) are partially occluded by the pyramid model and some of the image lines, e.g. line (2) are not formed by the projection of 3D edges. This extraneous portion of line (2) is due to an effect of shadowing. All these poor quality image lines were involved in the pose determination to examine the robustness of the proposed line-based method.

For each model and scene triplet correspondence MSTC (1) to (5), the geometric constraint equations were derived using the precomputed model triplet angle set and the corresponding image triplet attribute, as detailed in Section 6.5. In the case of the trihedral vertex, MSTC (5), the orientations of the image lines were computed at the canonical position where the optical axis is pointing to the tip of the vertex. The coefficients of the geometric constraints equations derived for each MSTC are given in Figure 6-14. The solutions of the constraint angle set $A^6$ measured at their respective canonical positions are tabulated in Figure 6-15 (A). The corresponding hypothesized poses computed using the formulation described in Section 6.7 are tabulated in Figure 6-15 (B).

Representative implausible hypotheses of MSTC (1), (3), (4) and (5) of pose solutions labelled as (○) in Figure 6-15 (B) are shown in Figures 6-16 (c), (d), Figures 6-17 (a) and (b), respectively. Having transformed the model to a predicted scene position, the confidence measure of each hypothesis was computed. In the case of MSTC (2), the perspective projective images of all the implausible hypotheses were at least 50% out of the image plane.

From the pose estimation results, we observed that the MSTC (1), (2), (3) from the hexagonal polyhedron and MSTC$_{i=4,5}$ from the pyramid model produce model-scene transform values that are relatively close to each other in their respective rotation and translation transformation space. These consensus transformations computed from the hexagonal polyhedron and pyramid model are labelled as (*) and (**) respectively (see Figure 6-15 (B)). Their superimposed images are shown in Figure 6-17 (c).

It is interesting to note that the orientation of the perspective image lines of the plausible hy-
Analysis of a triple of spatial edges

Hypothesized triplets of 3D edges involved in the pose determination align with the corresponding scene line, whereas the rest of the orientations of the lines are not so accurate. The similar observation was made in the experiment reported by Thompson and Mundy's [65] in the case of matching a vertex-pair. We believe this will not degrade the performance of the recognition system based on triplets of 3D edges. From the matching results obtained in the previous experiments, we found that the rotation and translation errors were relatively small.

In the Hough clustering approach, a significant cluster will still be detected as long as a reasonable number of correct model-scene triplet assignments are employed in the clustering process. The final pose of the scene object can be then refined by averaging the transformations of the model-scene triplet assignments which contribute to the peak in the six dimensional Hough space. In the hypothesis-verification paradigm, the initial hypotheses will still be generated, although the poses estimated from these hypothesized candidates may not be extremely accurate. In this regard, the results of the initial pose estimation can then be submitted to algorithms using an alignment approach of Lowe [50] or Huttenlocher and Ullman [40] for refinement.

In this example, the final pose of the hexagonal polyhedron and pyramid were determined by averaging the consensus transformations. The superimposed images of the target objects using the final estimated pose is shown in Figure 6-17 (d). These matching results show that the localization method has a reasonable accuracy in estimating the pose of the target objects.

From the consensus of the computed rotation and translation, we believe that triplets of relatively long image lines are most useful in the Hough-based approach. This conjecture will hold only if a reasonable number of plausible image triplets can be extracted from the scene image.

In general, connected image triplets and trihedral vertex-CS features can be reliably and robustly extracted from the scene image acquired in a noisy environment. We accept that under such noisy condition, the extraction of plausible image triplets will be accomplished without barring spurious image triplets. However, extraneous triplets will only increase the computation load of the system, but will not significantly increase the failure rate of the recognition system.

In the next experiment, triplets of connected image lines and trihedral vertex-CS features will be exploited to demonstrate their effectiveness and robustness in object recognition without assuming apriori knowledge of the model-image triplet correspondences. The matching result will then provide a focus for discussing the capability and feasibility of the recognition system based on these primitives.
Analysis of a triple of spatial edges

![Image of hexagonal polyhedral model and pyramid model]

**Figure 6-12:** A hexagonal polyhedron model and a pyramid model

<table>
<thead>
<tr>
<th>Line</th>
<th>End points of image line</th>
<th>(A)</th>
<th>(B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(-25,86) - (-75,53)</td>
<td>(3,2)</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>(-55,89) - (59,75)</td>
<td>(3,4)</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>(68,63) - (78,35)</td>
<td>(4,5)</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>(-78,19) - (-49,-10)</td>
<td>(8,7)</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>(-79,25) - (-78,48)</td>
<td>(8,2)</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>(-47, -12) - (-14, -14)</td>
<td>(7,6)</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>(79,5) - (53, -5)</td>
<td>-</td>
<td>(11,6)</td>
</tr>
<tr>
<td>8</td>
<td>(60,43) - (-27,53)</td>
<td>-</td>
<td>(10,9)</td>
</tr>
<tr>
<td>9</td>
<td>(20, -63) - (50, -7)</td>
<td>-</td>
<td>(0,2)</td>
</tr>
<tr>
<td>10</td>
<td>(16, -60) - (-2,24)</td>
<td>-</td>
<td>(0,3)</td>
</tr>
<tr>
<td>11</td>
<td>(-5,26) - (-18,0)</td>
<td>-</td>
<td>(3,1)</td>
</tr>
<tr>
<td>12</td>
<td>(17, -65) - (-19,1)</td>
<td>-</td>
<td>(0,1)</td>
</tr>
</tbody>
</table>

**Figure 6-13:** Extracted image lines and their corresponding model edges
Analysis of a triple of spatial edges

<table>
<thead>
<tr>
<th>MSTC</th>
<th>Constraint polynomial equation</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( (680.2826, -763.7564, 471.2869, -574.8115, 6.8778, 64.1326, 19.2112, 81.9970, 24.6694) )</td>
<td>( \tan \theta_1 )</td>
</tr>
<tr>
<td>2</td>
<td>( (-0.33936, 0.06772, 0.20888, -0.04724, 0.00215) )</td>
<td>( \tan \theta_1 )</td>
</tr>
<tr>
<td>3</td>
<td>( (55.5828, -43.0635, -474.5146, 230.1952, 991.3315, -171.6744, 184.7816, -15.7631, 7.4190) )</td>
<td>( \tan \theta_1 )</td>
</tr>
<tr>
<td>4</td>
<td>( (263.1217, -161.4429, 628.0559, -396.5838, 469.9711, -311.7409, 123.5158, -76.9446, 18.6085) )</td>
<td>( \tan \theta_1 )</td>
</tr>
<tr>
<td>5</td>
<td>( (0.64855, -3.84422, 7.58911, -7.74806, 6.76577) )</td>
<td>( 2 \left( \frac{1}{\sin \theta_1} - 1 \right) )</td>
</tr>
</tbody>
</table>

Figure 6-14: Geometric Constraint Equations

<table>
<thead>
<tr>
<th>MSTC</th>
<th>Constraint (A)</th>
<th>Constraint (B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( (\theta_1, \theta_2, \theta_3) )</td>
<td>( \Phi^e = (\Phi_x, \Phi_y, \Phi_z) )</td>
</tr>
<tr>
<td></td>
<td>(110.322, 12.578, 60.823)</td>
<td>(101.896, 103.002, 45.897)</td>
</tr>
<tr>
<td></td>
<td>(110.322, 117.382, 139.399)</td>
<td>(142.477, 337.703, 169.157)</td>
</tr>
<tr>
<td></td>
<td>(46.67, 83.90, 30.32)</td>
<td>(69.792, 169.387, 355.427)</td>
</tr>
<tr>
<td>2</td>
<td>( (\theta_1, \theta_2, \theta_3) )</td>
<td>( \Phi^e = (\Phi_x, \Phi_y, \Phi_z) )</td>
</tr>
<tr>
<td></td>
<td>(141.344, 117.512, 91.509)</td>
<td>(30.688, 304.287, 45.540)</td>
</tr>
<tr>
<td></td>
<td>(37.791, 63.211, 87.836)</td>
<td>(65.048, 166.998, 356.323)</td>
</tr>
<tr>
<td></td>
<td>(9.098, 84.048, 66.843)</td>
<td>(85.372, 146.020, 355.480)</td>
</tr>
<tr>
<td></td>
<td>(3.657, 87.617, 57.170)</td>
<td>(89.579, 141.840, 355.136)</td>
</tr>
<tr>
<td>3</td>
<td>( (\theta_1, \theta_2, \theta_3) )</td>
<td>( \Phi^e = (\Phi_x, \Phi_y, \Phi_z) )</td>
</tr>
<tr>
<td></td>
<td>(153.063, 124.242, 78.361)</td>
<td>(120.435, 329.297, 170.093)</td>
</tr>
<tr>
<td></td>
<td>(23.843, 56.766, 119.078)</td>
<td>(67.457, 170.256, 354.970)</td>
</tr>
</tbody>
</table>

Figure 6-15: The constraint angle solutions and estimated object’s poses
Analysis of a triple of spatial edges

Figure 6-16: The experimental results using a real image
Analysis of a triple of spatial edges

(a) Representative hypothesis of MSTC₄  
(b) Representative hypothesis of MSTC₅  
(c) The best hypotheses  
(d) The final match

Figure 6.17: The best hypotheses and the final match
Analysis of a triple of spatial edges

In this experiment, a widget with straight and curve edges was used to test the effectiveness of the proposed method based on line direction. Triplets of connected lines and trihedral vertex-CS features will be exploited to demonstrate their effectiveness and robustness in object recognition without assuming apriori knowledge of the model-image triplet correspondences.

A simplified version of the widget model is shown in Figure 6-20 (a). The planar curves of the widget are not employed as a key feature for generating hypotheses, but are involved in the verification process. This reflects the fact that, there is no distinct vertex at which the straight line and curve edge meet. This will clearly pose a serious problem for recognition method based on detection of accurate point features.

In this instance, there are 12 generic triplet groups \( M_{i=1,...,12} \) generated from the widget model. Figure 6-20 (b) shows the representative of each generic triplet groups. Each of the generic triplet group is a set of model triplet features \( M_i = \{ m_{i1}, m_{i2}, ..., m_{in_i} \} \), where \( m_{ij} \) is a model triplet represented by vector equations measured with respect to the model coordinate system and \( n_i \) is the number of model triples in the group. The number of model triples and trihedral vertices being classified in each group is shown in the top row of Figure 6-18 and Figure 6-19 respectively. The model triplet features of each generic triple group are characterised by a model triplet angle set \( A^\alpha \).

For example, the generic triplet group \( M_3 \) is attributed by \( A^\alpha = (\frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}) \).

In practice, due to measurement error the 3D space edge angles will not be exactly equal. Hence, two connected triplet features will be classified to the same generic triplet groups, if they are close within an allowable tolerance (1° was taken in the experiment). There are many model triples classified under the generic groups \( M_1 \) and \( M_3 \) corresponding to the coplanar rectangular bar end and orthogonal configurations respectively. These triple features are very common in practice.

Figure 6-21 (a) and (b) shows the grey-level image and lines extracted by Hough process, respectively. In this test image, 3 S-shaped image triple features \( F_{i=1,2,3} \) and 9 S-shaped image triple features \( F_{i=4,...,12} \) were identified from the grouping process. The extracted lines forming the key features are tabulated in the second column of Figure 6-18. In the case of the trihedral vertex, 4 such features \( F_{i=13,...,16} \) were identified in the scene image (see Figure 6-19). It is worth noting that both the true and spurious features such as \( F_6, F_8 \) are employed as key features for generating hypotheses in the matching process.

As mentioned in Section 6.10.3, invalid correspondences were not considered in the hypothesis and verification process. In this example, a representative invalid correspondence \( IC \) is the S-shaped image triplet \( F_1 \) and coplanar C-shaped triplet group \( M_1 \).

As mentioned in Section 6.9, for a given image triple and a generic triple group, there were two geometric constraint equations derived from two potential matches. The number of derived geometric constraint equations does not depend on the number of triple features in the group. For example, there were only two constraint equations derived from matching an image triple \( F_1 \) and a generic triple group \( M_6 \). The number of constraint equations to be derived would not depend on the number of triple features (in this case 4) in the generic triple group \( M_4 \). Given an image triple, some generic
Analysis of a triple of spatial edges

groups such as $M^*_I$ of symmetric model triple angle set $F^*_a = (\frac{\pi}{2}, 0, \frac{\pi}{2})$ are required to consider only once in deriving a constraint equation. These generic triple groups are labelled with (*) in the first row of Figure 6-18. In the case of an orthogonal trihedral vertex group $M^*_I$, only one geometric constraint equation was derived for considering each image vertex feature (see Figure 6-19).

The numbers of geometric constraints derived by matching model triples and trihedral vertices against all the extracted scene features are tabulated in the row NOGCE of Figure 6-18 and Figure 6-19, respectively. Having solved the triple geometric constraint equations, the number of constraint angle solutions determined by matching model triples and trihedral vertices against the scene features $F_{i=1,\ldots,16}$ are tabulated in the row TNS of Figure 6-18 and Figure 6-19, respectively. In some cases, there were no real solutions found when establishing the geometric relationships between image features and the generic triplet group.

These are labelled as NS in Figure 6-18 and 6-19. In this example, 148 geometric constraint equations were derived and solved for each valid image feature $F_i$ and generic triplet group $M_j$ correspondence.

For each solution set $\theta_{i=1,2,3}$ computed from a scene feature and a generic triplet group correspondence, the relative rotations and translation of the hypothesized objects were computed. Figure 6-21 (c) and (d) show representative hypotheses HYPO #1 and #2 generated by matching model triples against noise. Representative hypotheses HYPO #3 and #4 generated by matching trihedral model against wrong features are shown in Figure 6-21 (e) and (f). The hypotheses HYPO #5 and #6 shown in Figure 6-22 (a) and (b) are computed by matching the model against the correct feature but with wrong edge correspondences. Using the correct model and image triple correspondence, the hypothesis HYPO #7 of inaccurate pose estimation is shown in Figure 6-22 (c). However, the ones of accurate pose estimation were also generated. In the case matching model triples against the noise HYPO #8 and wrong features HYPO #9, #10 are shown in Figure 6-22 (d) and Figure 6-22 (e) (f), respectively. Figure 6-23 (a) shows a poor match generated by matching a correct corresponding feature. However, the right ones were also generated. In this example, 1370 feasible hypotheses were generated by considering all the model triple features and image triple exhaustively. At this stage, the remaining feasible hypotheses were then tested with the rules stated in Section 6.10.4. After applying the rules, the number of hypotheses for the widget model was reduced by 35.4%.

The confidence measure of each residual geometrically admissible hypothesis was computed. Figure 6-23 (e) shows the plot of the confidence measure against each plausible hypothesis. There was a hypothesis HYPO #12 shown in Figure 6-23 (b) of relatively low confidence (45.2%) measure obtained by matching the correct model triplet (in $M_1$) and image feature $F_{12}$ correspondence. We believe that the orientation of the coplanar rectangular bar end was relatively sensitive to the orientation of the image feature. Furthermore, all the lengths of the image segments $(1, 2, 10)$ forming the bar end were relatively short, and therefore their orientation is prone to errors. The wrong match did not degrade the capability of our recognition system, as there was a relatively large number of correct matches in the list of feasible hypotheses.
Analysis of a triple of spatial edges

A maximum confidence measure $P_8 = 71.11\%$ was identified from an entire list of feasible hypotheses. There were 11 hypotheses of confidence measure exceeding the threshold $0.9 \times P_8$. This cutoff threshold value is depicted by a dashed line shown in Figure 6-23 (e). The superimposed images of these high confidence hypotheses are shown in Figure 6-23 (c). The pose of the widget with respect to the camera frame was completed by averaging the translation and rotation transforms of the plausible hypotheses. The widget was successfully identified from the scene. The proposed pose determination yielded a reasonably good result. This can be easily verified by the closeness of the superimposed image shown in Figure 6-23 (d).

![Figure 6-18](image.png)

**Figure 6-18:** The image connected triples and the number of constraint angle sets

![Figure 6-19](image.png)

**Figure 6-19:** The image trihedral vertices and the number of constraint angle sets
Analysis of a triple of spatial edges

Figure 6-20: The description of widget model

(a) A simplified version of the CAD widget model

(b) Representative generic triple groups
Analysis of a triple of spatial edges

(a) test image

(b) A line drawing

(c) HYPO #1: (A, B, C) - (17, 7, 23)

(d) HYPO #2: (A, B, C) - (5, 6, 21)

(e) HYPO #3: (A, B, C) - (19, 20, 22)

(f) HYPO #4: (A, B, C) - (22, 16, 15)

Figure 6-21: Representative Hypotheses
Analysis of a triple of spatial edges

(a) HYPO #5 : (A, B, C) - (19, 20, 16)

(b) HYPO #6 : (A, B, C) - (20, 16, 15)

(c) HYPO #7 : (A, B, C) - (19, 20, 22)

(d) HYPO #8 : (A, B, C) - (4, 18, 3)

(e) HYPO #9 : (A, B, C) - (22, 16, 20)

(f) HYPO #10 : (A, B, C) - (9, 1, 2)

Figure 6-22: Representative Hypotheses
Analysis of a triple of spatial edges

(a) HYPO #11: (A, B, C) - (12, 16, 15)

(b) HYPO #12: (A, B, C) - (1, 2, 10)

(c) The best hypotheses

(d) The matching result

(e) A confidence measure plot

Figure 6-23: Representative hypotheses, best hypotheses and matching result
The aim of this experiment was to demonstrate the computational effectiveness and robustness of the recognition method for recognizing multiple scene objects in the presence of clutter and subject to partial occlusion. A hexagonal model (HEXM), a House model and a box model were employed in this experiment. The test scene contained the models surrounded and occluded by other objects (see Figure 6-27) (a). A white rectangular card was placed in the test scene. We believe this irrelevant feature will generate many infeasible hypotheses, as its shape is very similar to some of the model surfaces. This will then offer an opportunity to examine the capability of the verification process for pruning these implausible hypotheses.

In this example, there are only 8 generic triplet groups $M_i = t, \ldots, s$ precomputed from the three object models shown in Figure 6-24 (a). The geometric configurations of the triplet groups are shown in Figure 6-24 (b). The numbers of triplet and trihedral-vertex features of each model for each generic triplet group are given in Figure 6-25. It is interesting to note that the total number of generic triplet groups of the three models is less than the one generated from the widget model. This is because the three models share many common triplet features. We envisage that the computation overhead of deriving and solving the geometric constraint equations will be certainly less than those required for the previous experiment, provided the same number of image features are extracted from the scene.

Figure 6-27 (a) and (b) shows the grey-level image and lines extracted by Hough process, respectively. The Hough lines extracted from the scene were grouped to form image triples and trihedral vertices $F_i$ and are tabulated in Figure 6-26 (A1) and (B1) respectively. In this test scene, there were 14 S-shaped image triples, 38 C-shaped image triples and 9 trihedral vertices identified in the grouping process. Many spurious and irrelevant features such as $F_{i0}, F_{i3}, \ldots$ etc. were involved in the hypothesis generation phase.

As in the previous experiment, for each valid image feature $F_i$ and a generic triplet group $M_j$, two geometric constraint equations were derived for each valid image and model triplet pair. Generic triplet groups labelled with (*) were considered only once in deriving the geometric constraint equation. The numbers of constraint equations for matching the generic triple and trihedral groups against each image feature are tabulated in the row NOGCE in Figure 6-26. In this example, there were in total 316 geometric constraint equations derived from these correspondences. The number of real solution sets $A^\theta$ of the constraint angle solution sets is given in Figures 6-26 (A2) and (B2). The total number of constraint angle solutions is given in the row TNS in Figure 6-26. Two real and equal roots were counted as one root. The orientations of the triplet edges were determined using the computed constraint angle solution sets. Having recovered the edge orientations, the relative rotation and translation transforms were computed.

Representative hypotheses generated by matching the triples of the HEXM, house and box models against the spurious features are shown in Figures 6-27 (c), (d) and (e) respectively. Representative hypotheses generated by matching the HEXM, house and box models against the features of wrong models are shown in Figures 6-27 (f), Figures 6-28 (a) and (b) respectively. Figure 6-28 (c) and (d) show the representative hypothesized models for matching the trihedral vertices of the HEXM and
Analysis of a triple of spatial edges

house model against the spurious features. Representative hypotheses generated for matching the trihedral vertices of the box against the image triples of wrong models are shown in Figure 6-28 (e). The model and image triple edges of these hypotheses are given in Figure 6-26 (C).

In this example, there were 7872, 5548 and 3024 hypotheses generated by matching the image triples against the triple features of the hexagonal model (HEXM), the House model and the box model, respectively. After apply the rules stated in Section 6.10.4, the number of hypotheses generated from the (HEXM), the House model and the box model were reduced by 55.1 %, 48.5 % and 41.9 % respectively.

Having rejected the implausible hypotheses, the confidence measure of each residual geometrically admissible hypothesis was computed. Figure 6-30 (a), (b) and (c) show the plot of the confidence measure against each plausible hypothesis generated from the (HEXM), the House model and the box model respectively. The peaks $P_g$ identified in each plot were 70.8 %, 83.2 % and 66.5 %. The number of plausible hypotheses of confidence measures exceeding the cutoff threshold ($0.9 \times P_g$) which were identified from each confidence measure plot were 96, 16 and 32. Representative plausible hypotheses fell below the cutoff threshold are shown in Figure 6-28 (f). For each plot, the plausible hypotheses of rotation and translation transforms close to the ones associated with the peaks were identified. There were 9, 8 and 4 such hypotheses shown in Figure 6-30 (a), (b) and (c), respectively. The superimposed images of these best hypotheses are shown in Figure 6-29 (a). The final results shown in Figure 6-29 (b) were determined by averaging the transforms of the best hypotheses.

In this experiment, the three target objects were successfully identified and located from the scene image using the proposed matching strategies. The confidence measures associated with the correct hypotheses were very distinctive and significant.
Analysis of a triple of spatial edges

All the dimensions are given in (mm)

(a) The HEXM, house and box models

(b) Representative generic triple group

Figure 6-24: The descriptions of the HEXM, house and box models

<table>
<thead>
<tr>
<th></th>
<th>( M^*_1 )</th>
<th>( M^*_2 )</th>
<th>( M^*_3 )</th>
<th>( M^*_4 )</th>
<th>( M^*_5 )</th>
<th>( M^*_6 )</th>
<th>( M^*_7 )</th>
<th>( M^*_8 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEXM</td>
<td>24</td>
<td>12</td>
<td>-</td>
<td>-</td>
<td>24</td>
<td>12</td>
<td>-</td>
<td>12</td>
</tr>
<tr>
<td>House</td>
<td>22</td>
<td>4</td>
<td>4</td>
<td>12</td>
<td>12</td>
<td>6</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>Box</td>
<td>24</td>
<td>-</td>
<td>-</td>
<td>24</td>
<td>-</td>
<td>-</td>
<td>8</td>
<td>-</td>
</tr>
<tr>
<td>Total</td>
<td>70</td>
<td>16</td>
<td>4</td>
<td>36</td>
<td>36</td>
<td>18</td>
<td>12</td>
<td>18</td>
</tr>
</tbody>
</table>

Figure 6-25: The descriptions of generic triple groups of the 3 models
Analysis of a triple of spatial edges

<table>
<thead>
<tr>
<th>(A1)</th>
<th>(A2)</th>
<th>(B1)</th>
<th>(B2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>Edge Labels</td>
<td>$M^+$</td>
<td>$M^-$</td>
</tr>
<tr>
<td>$F_1$</td>
<td>(4, 14, 26)</td>
<td>IC</td>
<td>IC</td>
</tr>
<tr>
<td>$F_2$</td>
<td>(6, 30, 27)</td>
<td>IC</td>
<td>IC</td>
</tr>
<tr>
<td>$F_3$</td>
<td>(13, 6, 23)</td>
<td>IC</td>
<td>IC</td>
</tr>
<tr>
<td>$F_4$</td>
<td>(1, 25, 29)</td>
<td>IC</td>
<td>IC</td>
</tr>
<tr>
<td>$F_5$</td>
<td>(22, 23, 6)</td>
<td>IC</td>
<td>IC</td>
</tr>
<tr>
<td>$F_6$</td>
<td>(27, 14, 8)</td>
<td>IC</td>
<td>IC</td>
</tr>
<tr>
<td>$F_7$</td>
<td>(17, 21, 39)</td>
<td>IC</td>
<td>IC</td>
</tr>
<tr>
<td>$F_8$</td>
<td>(11, 16, 21)</td>
<td>IC</td>
<td>IC</td>
</tr>
<tr>
<td>$F_9$</td>
<td>(19, 21, 16)</td>
<td>IC</td>
<td>IC</td>
</tr>
<tr>
<td>$F_{10}$</td>
<td>(15, 8, 31)</td>
<td>IC</td>
<td>IC</td>
</tr>
<tr>
<td>$F_{11}$</td>
<td>(27, 14, 31)</td>
<td>IC</td>
<td>IC</td>
</tr>
<tr>
<td>$F_{12}$</td>
<td>(9, 10, 2)</td>
<td>IC</td>
<td>IC</td>
</tr>
<tr>
<td>$F_{13}$</td>
<td>(12, 16, 21)</td>
<td>IC</td>
<td>IC</td>
</tr>
<tr>
<td>$F_{14}$</td>
<td>(23, 6, 30)</td>
<td>IC</td>
<td>IC</td>
</tr>
<tr>
<td>$F_{15}$</td>
<td>(4, 14, 8)</td>
<td>I</td>
<td>NS</td>
</tr>
<tr>
<td>$F_{16}$</td>
<td>(12, 16, 17)</td>
<td>I</td>
<td>2</td>
</tr>
<tr>
<td>$F_{17}$</td>
<td>(16, 21, 39)</td>
<td>I</td>
<td>1</td>
</tr>
<tr>
<td>$F_{18}$</td>
<td>(17, 21, 19)</td>
<td>I</td>
<td>NS</td>
</tr>
<tr>
<td>$F_{19}$</td>
<td>(36, 5, 18)</td>
<td>I</td>
<td>2</td>
</tr>
<tr>
<td>$F_{20}$</td>
<td>(13, 30, 27)</td>
<td>I</td>
<td>2</td>
</tr>
<tr>
<td>$F_{21}$</td>
<td>(1, 25, 22)</td>
<td>I</td>
<td>NS</td>
</tr>
<tr>
<td>$F_{22}$</td>
<td>(23, 22, 29)</td>
<td>I</td>
<td>2</td>
</tr>
<tr>
<td>$F_{23}$</td>
<td>(23, 22, 25)</td>
<td>I</td>
<td>1</td>
</tr>
<tr>
<td>$F_{24}$</td>
<td>(25, 29, 28)</td>
<td>I</td>
<td>1</td>
</tr>
<tr>
<td>$F_{25}$</td>
<td>(28, 35, 24)</td>
<td>I</td>
<td>4</td>
</tr>
<tr>
<td>$F_{26}$</td>
<td>(18, 19, 39)</td>
<td>I</td>
<td>2</td>
</tr>
<tr>
<td>$F_{27}$</td>
<td>(14, 8, 15)</td>
<td>I</td>
<td>2</td>
</tr>
<tr>
<td>$F_{28}$</td>
<td>(8, 15, 20)</td>
<td>I</td>
<td>3</td>
</tr>
<tr>
<td>$F_{29}$</td>
<td>(18, 19, 21)</td>
<td>I</td>
<td>2</td>
</tr>
<tr>
<td>$F_{30}$</td>
<td>(4, 14, 31)</td>
<td>I</td>
<td>4</td>
</tr>
<tr>
<td>$F_{31}$</td>
<td>(26, 14, 27)</td>
<td>I</td>
<td>3</td>
</tr>
<tr>
<td>$F_{32}$</td>
<td>(11, 16, 17)</td>
<td>I</td>
<td>2</td>
</tr>
<tr>
<td>$F_{33}$</td>
<td>(24, 26, 14)</td>
<td>I</td>
<td>4</td>
</tr>
<tr>
<td>$F_{34}$</td>
<td>(29, 28, 35)</td>
<td>I</td>
<td>2</td>
</tr>
<tr>
<td>$F_{35}$</td>
<td>(5, 18, 19)</td>
<td>I</td>
<td>2</td>
</tr>
<tr>
<td>$F_{36}$</td>
<td>(24, 26, 8)</td>
<td>I</td>
<td>2</td>
</tr>
<tr>
<td>$F_{37}$</td>
<td>(4, 27, 30)</td>
<td>I</td>
<td>2</td>
</tr>
<tr>
<td>$F_{38}$</td>
<td>(32, 36, 34)</td>
<td>I</td>
<td>1</td>
</tr>
<tr>
<td>$F_{39}$</td>
<td>(36, 34, 33)</td>
<td>I</td>
<td>1</td>
</tr>
<tr>
<td>$F_{40}$</td>
<td>(34, 33, 32)</td>
<td>I</td>
<td>NS</td>
</tr>
<tr>
<td>$F_{41}$</td>
<td>(33, 32, 36)</td>
<td>I</td>
<td>2</td>
</tr>
<tr>
<td>$F_{42}$</td>
<td>(26, 8, 15)</td>
<td>I</td>
<td>2</td>
</tr>
<tr>
<td>NOGCE</td>
<td>28</td>
<td>28</td>
<td>56</td>
</tr>
<tr>
<td>TNS</td>
<td>28</td>
<td>51</td>
<td>104</td>
</tr>
</tbody>
</table>

Figure 6-26: The extracted scene image features and the constraint angle solutions
Analysis of a triple of spatial edges

Figure 6-27: A test scene image and infeasible hypotheses
Analysis of a triple of spatial edges

Figure 6-28: Infeasible hypotheses
Analysis of a triple of spatial edges

(a) The best hypotheses

(b) The final matching results

Figure 6-29: The best hypotheses and the correct matches
Analysis of a triple of spatial edges

(a) The confidence measure plot of the HEXA model hypotheses

(b) The confidence measure plot of the house model hypotheses

(c) The confidence measure plot of the box model hypotheses

Figure 6-30: The matching results from the real image involving multiple target objects
Analysis of a triple of spatial edges

6.12 Conclusion

An intuitive mathematical formulation has been proposed for the interpretation of the geometric relationships of a triple of spatial edges and their perspective projection forming image lines. No restriction is imposed on the configuration of the triple of spatial edges. An eighth-degree polynomial equation explicitly defined by the space angles between the corresponding three spatial edges measured with respect to an object centered coordinate system has been derived. The crux of this representation is that the angular attributes of pairs of spatial edges are object-independent. There are two important aspects of using this novel representation:

- An important aspect of the use of angular representation of triples is that parts of whole object models can be decomposed or expressed using triples as basis primitives. This means that only a few generic types of triple have to be defined and can then be used to describe a wide range of object models.

- The computational load depends on the number of distinct groups of triples. It avoids replicating the same recognition module for every occurrence of the triple feature in the same generic triple group. Each group are distinguished by the angles between the constituent model edges. Many objects will contain triples which fall into existing classes and this implies there will be no increase in computation when there are added to the model database. Hence computation can be sublinear in terms of the number of models in the database.

An effective hypothesis generation scheme has been proposed which can take advantage of the commonality of this novel representation.

Particular closed form solutions have been derived for specific but common configurations of edges such as rectangular bar ends and orthogonal trihedral vertex. The practical significance and generality of our results are multifold. Extensive experiments have been performed to verify the plausibility of employing connected triple edges and trihedral vertices as key features in the paradigm of hypothesis-generation and Hough-clustering approaches to object recognition.

In general, the success rate for identifying and localising polyhedral objects using the proposed recognition method is relatively high. The poses of the target objects estimated using triples of connected edges and trihedral vertices are reasonably good.
Chapter 7

Discussion and Conclusion

7.1 Summary of Work

The issues and problems of modelling, feature extraction and polyhedral object recognition have been addressed. The methods and paradigms for solving these problems have been proposed and developed.

The analysis of Generalized cylinders has been carried out and presented in Chapter 3. A number of useful expressions and properties of the contour generators of SHGCs have been derived and discussed. Their 2D projected contours are analysed under perspective projection. The right and oblique straight homogeneous generalized cylinders (SHGC's) with circular and arbitrary cross-section have been considered. The characteristics and performance of a preliminary version of an object modelling system have been described and discussed. Extensions of the modelling system have been proposed and discussed with respect to object model invocation and geometric matching.

In Chapter 4, a new algorithm for grouping 2D line segments into open and closed polygons that correspond to feasible physical 3D structures has been presented. The algorithm starts by identifying junctions made of two line segments and then forms triples by combining pairs of junctions which share a common line. These triples are then scanned by a procedure which connects them into polygon structures. Heuristic rules are used to control the combinatorial explosion associated with unconstrained associations of junctions and triples. Physical rules are used to reject polygons which are incompatible with a single planar surface hypothesis. The algorithm does not require strict connectivity of end-points at junctions. The feature is seen as an intermediate level grouping which can be employed to extract features such as triplets of connected image lines, triangle-pairings, image trihedral vertex and closed polygons.

In Chapter 5, the problem of model based recognition of polyhedral objects from a single perspective view has been considered. A hypothesize-verify paradigm based on the use of high level knowledge constraints derived from local shape properties has been presented. In the recognition system, two high-level features, namely triangle-pair and quadrilateral are employed as key features for model invocation and hypothesis generation. A verification process for performing a detailed
check on the model-to-scene correspondences has been developed. To reduce the number of implausible hypotheses generated from scene-to-model intermediate feature assignments, two geometrical constraints, namely distance and angle constraints, have been proposed. A list of closed polygons and connected triples extracted from a 2D intensity image by means of edge and intermediate feature detection process described in Chapter 4 is used as input to the matching system. These triples are then scanned by a procedure which connects them into C-triple pair features. As a by-product of the recognition method the relative pose of the 3D polyhedral objects with respect to the camera is recovered. Detailed experimental results are reported to confirm the feasibility of the proposed method.

In Chapter 6, an intuitive mathematical formulation has been proposed for the interpretation of the geometric relationships of a triple of spatial edges and their perspective projection forming image lines. No restriction is imposed on the configuration of the triple of spatial edges. An eighth-degree polynomial equation explicitly defined by the space angles between the corresponding three spatial edges measured with respect to an object centered coordinate system has been derived. The crux of this representation is that the angular attributes of pairs of spatial edges are object-independent. There are two important aspects of using this novel representation:

- An important aspect of the use of angular representation of triples is that parts of whole object models can be decomposed or expressed using triples as basis primitives. This means that only a few generic types of triple have to be defined and can then be used to describe a wide range of object models.

- The computational load depends on the number of distinct groups of triples. It avoids replicating the same recognition module for every occurrence of the triple feature in the same generic triple group. Each group is distinguished by the angles between the constituent model edges. Many objects will contain triples which fall into existing classes and this implies there will be no increase in computation when there are added to the model database. Hence computation can be sublinear in terms of the number of models in the database.

An effective hypothesis generation scheme has been proposed which can take advantage of the commonality of this novel representation.

Particular closed form solutions have been derived for specific but common configurations of edges such as rectangular bar end and orthogonal triple. The practical significance and generality of our results are multifold. Extensive experiments have been performed to verify the plausibility of employing connected triple edges and trihedral vertices as key features in the paradigm of hypothesis-generation and Hough-clustering approaches to object recognition. The poses of the object estimated using triples of edges are reasonably good.
7.2 Problems

7.2.1 Missing Significant Image Features

The accuracy of the estimated object's poses and the robustness of the proposed recognition methods depend entirely on the extraction of geometric primitives such as triples of connected image lines, image trihedral vertices, triangle-pairings and closed polygons. Some of these seed features may however be missing due to occlusion or inadequate low-level processing.

To cope with these problems, the low confidence or poor quality features can be enhanced by modifying the thresholds on proximity and orientation checks. However, if the tolerance is too large, the number of spurious features extracted from a scene may cause the model-scene correspondences to grow exponentially. As a result, the computational load of the verification process will increase dramatically, degrading the overall efficiency of the system. This is one of the important issues which can be solved by implementing an adaptive control mechanism for providing an interactive environment between the matching phase and low-level or feature grouping process.

In the control framework, the matching phase should always communicate with the feature extraction modules. So that the thresholds employed in the grouping process can be increased or decreased gradually in accordance with the matching results and clues aggregated through the previous image interpretation experience.

7.2.2 Implausible Hypotheses yield High Confidence Measures

In general, the success rate for identifying and localising target objects using the proposed paradigm is relatively high. However, wrongly hypothesized models will occasionally yield a very high confidence measure if the projection of the hypothesized model is very small compared with respect to the image plane. This is because the probability of a small region to be fully occupied by a few irrelevant features is very high.

These hypotheses may be pruned away using a classical hidden line removal algorithm test. Even if the hidden line removal algorithm test is used the conclusion drawn from this test at this configuration is not reliable. This is because the relative distances between model vertices and the computed surface normals are very small as compared to the distance from the camera to the centroid of the object. Hence the computation involved in the test are highly unstable. We propose two possible methods to overcome this problem.

1. Using Global Consistency : The hypothesized objects must not protrude inside or be occluded by other hypothesized objects stored in the description of the scene interpretation. The spatial description of a scene can be constructed by aggregating the matching results derived over a period of time. Clearly, the assumption made here is that the object identified earlier are highly plausible.

If a priori knowledge of the bounds on the target object location are known, the problem can
Discussion and Conclusion

be greatly simplified. Stability is one of the useful physical constraints of a rigid object which can be exploited as a priori knowledge. However, this might not be the case in general vision, where a robot is exploring a new environment.

2. Using Active Vision Paradigm: Many researchers have proposed to move the camera to positions where the analysis of the vision problems can be greatly simplified. Certainly, this provides one of the possible solutions to this problem. One can verify the predicted features in more detail by moving the camera closer to the hypothesized objects in small steps to avoid collision. However, this will be an effective solution only if the recognition system has a fairly good understanding about the environment by building a model of the scene.

7.3 Future Work

7.3.1 Organization of Model base

Currently, the creation of models and the organization of the geometric model features are performed manually. This is a very tedious and non-productive task. An automatic model generation process could be developed to derive and organize key features into useful data structures for matching purposes.

7.3.2 Improvement to the Verification Process

In our framework of polyhedral object recognition, the verification process is performed by correlating the predicted projected 2D geometric primitives such as line segments and angles between segments with the extracted 2D scene features. Lowe [50] proposed to use object surface properties such as color, shading and texture measures for final verification. The evaluation of these measures is generally computationally intensive. Hence, we will select only those hypotheses whose confidence measure exceeds a certain threshold for further verification using such properties. A rigorous model could be developed to combine confidence measures computed from 2D geometric features and surface properties. One should think of whether it is more effective to combine the confidence measures of each knowledge source one at time or all at once. In the former approach, the verification process can be terminated when a hypothesis of sufficiently high confidence measure has been identified.

7.3.3 Extensions for Recognition of Curved Objects

The feasibility of the matching strategies described in this dissertation has been demonstrated within the domain of polyhedral objects. Some of the model invocation techniques, hypothesis generation schemes and formulations designed for polyhedral object recognition can be naturally extended to recognizing curved objects.

Three main issues will be considered in future: the organisation of GC model descriptions into
Discussion and Conclusion

a hierarchical structure; the development of strategies and techniques for model invocation based on GC primitives; the implementation of symbolic and geometric model-based matching using both 2D and 3D features derived from the GC modelling system.

The object model descriptions will be organised into symbolic (or qualitative) and geometric (or quantitative) levels. In the symbolic level, view-point invariant, salient and distinctive features can be stored and used for model invocation thus reducing the search space in matching. At the quantitative description level, geometric features such as contour generators, 2D projected contours, curvatures and junction types formed between contours and creases could be evaluated and used for both 2D and 3D geometric matching.
Appendix A

The transform \( \mathcal{R}(P,f) \) is the standard rotation transformation which maps the image point \( P = (P_x, P_y) \) onto the image origin \((0,0)\).

\[
\mathcal{R}(P,f) = \begin{pmatrix}
E & F & l_1 \\
F & G & l_2 \\
-l_1 & -l_2 & l_3
\end{pmatrix}
\]  

(A.1)

The corresponding image standard transformation \( T(P,f) \) induced by camera rotation \( \mathcal{R}(P,f) \) is defined as:

\[
\begin{align*}
\chi' &= f \frac{E x + F y - l_1 f}{l_1 x + l_2 y + l_3 f} \\
y' &= f \frac{F x + G y - l_2 f}{l_1 x + l_2 y + l_3 f}
\end{align*}
\]  

(A.2)

where \( l \) is the 3D unit vector emanating from the camera viewpoint pointing toward the image point \( P = (P_x, P_y) \) and is given by,

\[
l = (l_1, l_2, l_3) = \left( \frac{P_x}{\sqrt{x^2+y^2}}, \frac{P_y}{\sqrt{x^2+y^2}}, \frac{f}{\sqrt{x^2+y^2}} \right)
\]

and the \( E, F \) and \( G \) are functions of the image point \( P \) and the focal length \( f \) of the camera. They can be expressed as,

\[
\begin{align*}
E &= \frac{P_x^2 l_3 + P_y^2}{\kappa} \\
F &= \frac{P_x P_y (l_3 - 1)}{\kappa} \\
G &= \frac{P_y^2 l_3 + P_x^2}{\kappa}
\end{align*}
\]

where \( \kappa = P_x^2 + P_y^2 \) is the module of the position vector of the image point \( P \) expressed in the origin of the image plane. For detailed derivations, the reader is referred to the interesting work by Kanatani [44].
Appendix B

Table B.1

| \(v_{11} = -\cos \alpha_{23} K_3(\phi_1)\) | \(v_{12} = -r_{33} \cos \alpha_{23}\) | \(v_{13} = \cos \alpha_{13} K_3(\phi_2)\) | \(v_{14} = r_{33} \cos \alpha_{13}\) |
| \(v_{21} = \cos \alpha_{23} K_1(\phi_1)\) | \(v_{22} = \cos \alpha_{23} K_2\) | \(v_{23} = -\cos \alpha_{13} K_1(\phi_2)\) | \(v_{24} = -\cos \alpha_{13} K_2\) |

\[v_{31} = \left((r_{12} r_{23} - r_{13} r_{22}) \sin \phi_3 + (r_{11} r_{23} - r_{13} r_{21}) \cos \phi_3\right) \left(\sin \phi_2 \cos \phi_1 - \cos \phi_2 \sin \phi_1\right)\]

\[= (r_{31} \sin \phi_3 - r_{32} \cos \phi_3) \sin(\phi_2 - \phi_1)\]

\[v_{32} = -K_4(\phi_2)\]

\[v_{33} = K_4(\phi_1)\]

\[K_4(\phi_{1,2}) = \left((r_{22} r_{33} - r_{23} r_{32}) \sin \phi_4 + (r_{12} r_{33} - r_{13} r_{32}) \cos \phi_4\right) \sin \phi_3\]

\[+\left((r_{11} r_{33} - r_{13} r_{31}) \cos \phi_4 + (r_{21} r_{33} - r_{23} r_{31}) \sin \phi_4\right) \cos \phi_3\]

\[= (r_{11} \sin \phi_4 - r_{12} \cos \phi_4) \sin \phi_3 + (r_{22} \cos \phi_4 - r_{12} \sin \phi_4) \cos \phi_3\]

Table B.2

\(\mu_0 = -v_{11}^2\)

\(\mu_1 = -v_{12}^2\)

\(\mu_2 = -v_{13}^2\)

\(\mu_3 = -2 v_{31} v_{32}\)

\(\mu_4 = -2 v_{31} v_{33}\)

\(\mu_5 = -2 v_{32} v_{33}\)

\(\mu_6 = v_{33}^2 + v_{13}^2\)

\(\mu_7 = v_{33}^2 + v_{11}^2\)

\(\mu_8 = v_{14}^2 + v_{14}^2\)

\(\mu_9 = v_{13}^2 + v_{12}^2\)

\(\mu_{10} = 2 (v_{21} v_{23} + v_{11} v_{13})\)

\(\mu_{11} = 2 (v_{23} v_{24} + v_{13} v_{14})\)

\(\mu_{12} = 2 (v_{22} v_{23} + v_{12} v_{13})\)

\(\mu_{13} = 2 (v_{24} v_{24} + v_{11} v_{14})\)

\(\mu_{14} = 2 (v_{21} v_{22} + v_{11} v_{12})\)

\(\mu_{15} = 2 (v_{22} v_{24} + v_{12} v_{14})\)

Table B.3

\(\sigma_{00} = (\mu_8 + \mu_9) - \mu_{15}\)

\(\sigma_{01} = 2 (\mu_{14} - \mu_{13})\)

\(\sigma_{02} = 2 (2 (\mu_2 + \mu_7) + \mu_8 - \mu_9)\)

\(\sigma_{03} = -2 (\mu_{13} + \mu_{14})\)

\(\sigma_{04} = (\mu_8 + \mu_9 + \mu_{15})\)

\(\sigma_{10} = 2 (\mu_{11} - \mu_{12})\)

\(\sigma_{11} = 4 (\mu_{10} - \mu_5)\)

\(\sigma_{12} = 4 (2 \mu_4 + \mu_{11})\)

\(\sigma_{13} = 4 (\mu_5 + \mu_{10})\)

\(\sigma_{14} = 2 (\mu_{11} + \mu_{12})\)

\(\sigma_{20} = 2 (2 (\mu_1 + \mu_6) - \mu_8 + \mu_9)\)

\(\sigma_{21} = -4 (2 \mu_3 + \mu_{14})\)

\(\sigma_{22} = 4 (4 \mu_0 - 2 (\mu_1 + \mu_2 - \mu_5 - \mu_7) - \mu_8 - \mu_9)\)
Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


