ESD-TR-88-194

Technical Report
817

Model-Based Silhouette Recognition

P.L. Van Hove
R.A. Jaenicke

25 November 1988

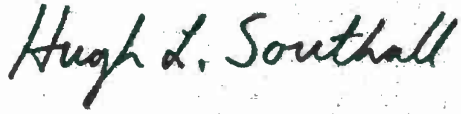
Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

Prepared for the Department of the Air Force
under Electronic Systems Division Contract F19628-85-C-0002.

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology, with the support of the Department of the Air Force under Contract F19628-85-C-0002.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER



Hugh L. Southall, Lt. Col., USAF
Chief, ESD Lincoln Laboratory Project Office

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

MODEL-BASED SILHOUETTE RECOGNITION

P.L. VAN HOVE
R.A. JAENICKE
Group 21

TECHNICAL REPORT 817

25 NOVEMBER 1988

LEXINGTON MASSACHUSETTS

ABSTRACT

We present a system for recognizing 3-D objects at unknown orientations from their 2-D silhouettes. The geometric description of an object model is provided in CAD form and is then compiled into a set of geometric constraints for a large set of viewing directions. The silhouette is parsed into a set of straight edges, and these edges are compared to the edges of the model by conceptually structuring all possible interpretations in a tree. This enormous search space is pruned by extending the interpretation tree search of Grimson and Lozano-Perez to work for the 3-D model/2-D data case. This includes a precise analysis of the propagation of errors in the position and orientation of silhouette edges, which then provide adequate constraints for pruning the search tree. Any hypotheses that survive the pairwise constraints of tree search are verified by synthesizing a silhouette of the model for the hypothesized orientation and comparing this synthetic silhouette to the observed silhouette.

Based only on silhouette data, the system can find all plausible interpretations of the data, including symmetric viewpoints. The system performs in the presence of unknown viewpoint, moderate scale uncertainties, occluding objects, and degradations in the silhouette shape due to image noise and image processing artifacts. These characteristics should enable the system to perform well in applications where images have reasonable spatial resolution but where limited resolution in the signal (intensity or range) reduces the information in the data to a silhouette.

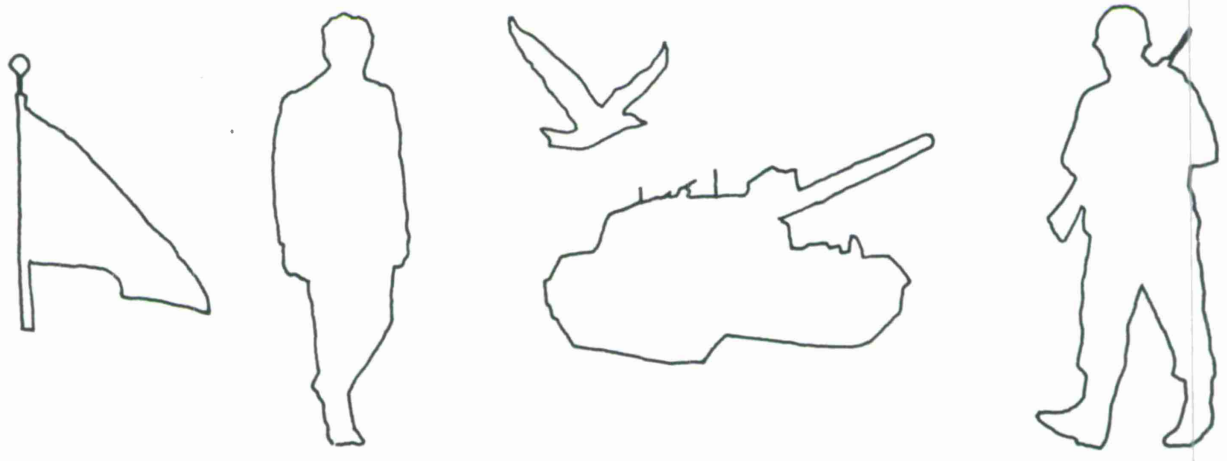
TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

1. INTRODUCTION

1.1 SILHOUETTES AND MODELS

Silhouettes represent an important class of object features. In many application domains, limits on resolution and signal-to-noise ratio reduce the usefulness of grayscale images to that of a binary image or a silhouette. While such domains severely limit the amount of information available for object recognition, human beings are easily capable of recognizing silhouettes of complex 3-D objects, such as those depicted in Fig. 1-1.



105340-1

Figure 1-1. Silhouettes of commonplace objects.

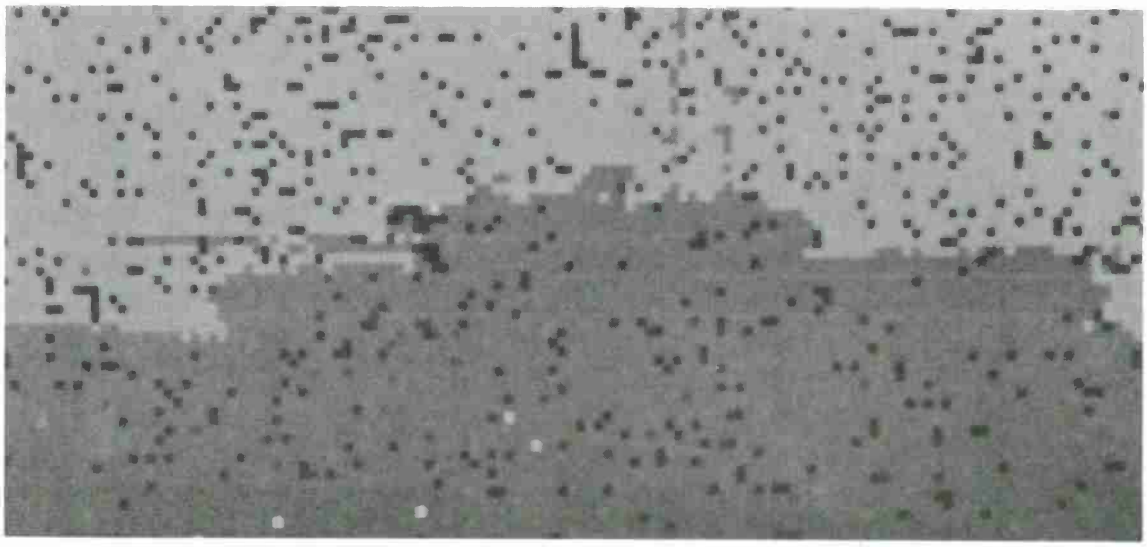
Recognition of silhouettes implies a knowledge of the objects that can produce silhouettes; therefore, we must have a set of models for the objects of interest in the scene. While most vision algorithms incorporate some model of the world, we call a recognition system "model-based" if there is a model for each object of interest that includes sufficient detail to permit recognition of that object. Most model-based recognition systems then consist of two distinct parts. The first is the *model-formation stage* in which object models are created off-line. The second is the *recognition stage* in which an instance of the model is located in the image in an on-line process.

Recognition of silhouettes also implies a knowledge of the process by which silhouettes are formed. We must have a projective transformation that can, to some approximation, create the instances of the objects in the sensor data. What we are looking for then is the inverse transformation to get us from the projection of the object in the sensor data back to the models. The problem is that the projective transformation is destructive. That is, information is lost during the transformation with the result that the object models contain more information than occurs in the the sensor data. Thus it is not possible to reconstruct a complete model from a single image and then perform the comparison at the model level. The relationship between objects and their silhouettes is discussed in further detail in Section 2.2.

1.2 THE RECOGNITION TASK

Most commercial vision systems currently available are restricted to the recognition of 2-D objects in 2-D images (2D/2D vision). Recognition of 3-D objects from 2-D images (3D/2D vision) is often achieved by exploiting a-priori information on the position and orientation of the objects of interest. Although these a-priori expectations can be justified for some applications, such as certain robotics problems, they largely reduce the generality of the vision systems. To give an example in silhouette recognition, if one assumes a known viewing direction, then there is only one possible shape for the silhouette, thereby reducing it from a 3D/2D vision problem to a 2D/2D problem.

Early failures at solving the 3D/2D vision problem have led researchers to develop more sophisticated sensors to produce full 3-D images of the object, providing fine detail of the relief. Although 3-D vision from 3-D images (3D/3D vision) is more straightforward than from 2-D images, the 3-D sensors are generally much more sophisticated and costly than their 2-D counterparts, and they are impractical in a large number of cases. In the example of Fig. 1-2, the image is provided by a 3-D sensor, but the resolution of the range measurements prevents their use in recognition so that the data is inherently two-dimensional. The range measurements would be-



105340-2

Figure 1-2. Gray-scale coded range image obtained with laser radar.

come useful in recognition if their resolution was increased by a factor of about 20; however, this would require an excessive increase in the signal/noise ratio. It is hence important to recognize 3-D objects with unknown orientations from the simpler 2-D images provided by a majority of sensors.

The recognition of 3-D objects from 2-D silhouette information is among the harder problems in machine vision. As mentioned previously, 3D/2D vision is an inverse transformation that is highly

ambiguous in the absence of a-priori information. Silhouette recognition is even more difficult, since silhouettes contain less information than do complete images that include inside details of the objects. A second difficulty is also common to all 3D/2D vision problems. Specifically, the imaging transformation has at least 5 degrees of freedom in this case, compared to 3 degrees of freedom for the transformation from a 2-D model to a 2-D image. The correspondence between a simple image feature and a model feature provides 2 constraints on the transformation. Therefore, the pairing of two image features with two model features is sufficient to determine a 2-D to 2-D transformation, perhaps up to a 2-fold ambiguity. However, at least three simple image features must be matched to determine a transformation from 3-D to 2-D, up to 2- or 4-fold ambiguities. The determination of the projection transformation is hence much more difficult for the recognition of 3-D objects than for the recognition of 2-D objects. In 3D/3D vision, the transformation has at least 6 degrees of freedom, but each simple image feature provides 3 constraints on the transformation so that the determination of the transformation is simpler in this case.

Determination of the transformation for a general projection of a 3-D object onto a 2-D image will allow for variations in the viewing direction and variations in scale caused by the viewing distance. Differences in scale between the model and the actual object must also be accounted for. To provide a robust solution to the 3D/2D vision problem, a recognition system must also perform well in the presence of noise and partial occlusions. Since one of the motivations for using silhouettes is to perform the recognition task when the sensor data is too noisy or of too low a resolution, clearly a silhouette recognition system should be able to handle noisy data gracefully. Partial occlusions also become important when objects can not be segmented easily on the basis of some simple criterion. In that case, a combined silhouette will be presented to the system to which a model will make only a partial match.

1.3 THE SILC SYSTEM

To perform the task of objection recognition from a single silhoutte image, we have designed the SILC software system. The SILC system compares a silhouette in the input image, taken with an unknown viewpoint, with a list of object models in its database and decides which model(s) and which viewpoint(s) correspond to the input data. A simple example of shapes that are identified by the computer is shown in Fig. 1-3; the silhouettes in the figure were easily identified by SILC, given numerical descriptions of the 3-D object models in Fig. 1-4.

The design of SILC was motivated by a problem of recognizing targets in range images produced by a laser radar developed at the Laboratory [12,28]. The example of a low-resolution range image shown earlier, Fig. 1-2, is a typical example of the type of data produced by the radar. It is apparent from this image that the most robust feature identifying the target in the image is its silhouette. There are numerous other applications where objects must be recognized based only on the shapes of their silhouettes, or where silhouettes provide the foremost identification cue among image features. These include images obtained with passive optical systems, with passive and active infrared sensors and with range/doppler radars.

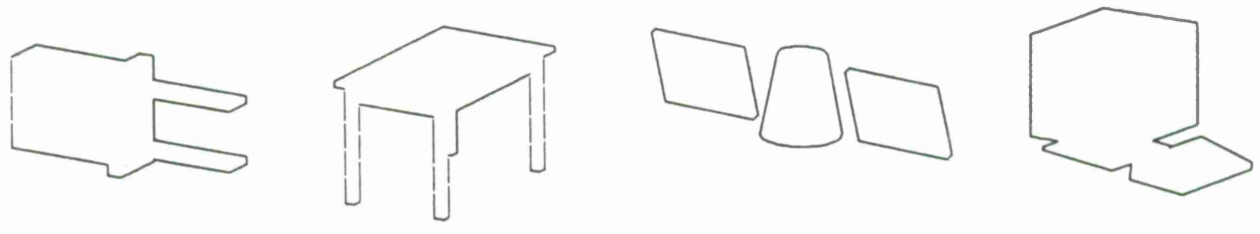


Figure 1-3. Silhouettes of simple geometric objects.

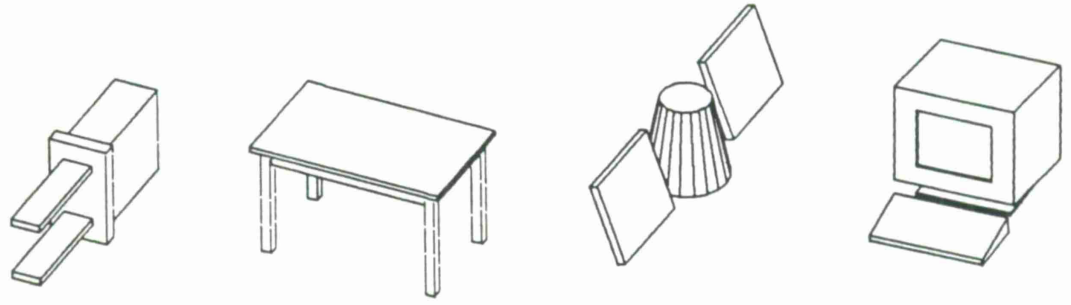


Figure 1-4. Models corresponding to the simple silhouettes.

The image in Fig. 1-2 also illustrates the need for the system to recognize the object (in this case a tank) from an unknown viewing direction. In the context of this application it must perform recognition for any rotation of the target around a vertical axis, for any target tilt due to terrain slopes, and for any sensor tilt in airborne data.

The experimental SILC system does recognize 3-D objects when presented with their silhouettes in images taken from unknown viewpoints. Although a number of other experimental systems have demonstrated recognition of 3-D objects given 2-D images from unknown viewpoints, their performance has been illustrated only by a few examples and has not been demonstrated in the context of silhouette identification. The SILC system is an extension of a state-of-the-art 2D/2D vision system developed by Grimson and Lozano-Perez [11] to the harder problem of 3D/2D vision. This system and others are discussed further in Section 2.1.

1.3.1 System Characteristics

The SILC system

- Recognizes 3-D Objects in 2-D Images.
- Bases the Recognition on Silhouette Data Only.
- Compares the Inputs with a Database of Polyhedral Object Models.
- Finds All Plausible Interpretations of the Data.

- Performs In the Presence of
    - Unknown Viewpoint.
    - Moderate Scale Uncertainties (20%).
    - Occlusions, resulting in Missing Features.
    - Superimposed Objects, resulting in Imperfect Segmentation.
    - Image Noise, resulting in Silhouette Shape Degradations.
    - Early Vision Artifacts, resulting in Spurious/Degraded Features.

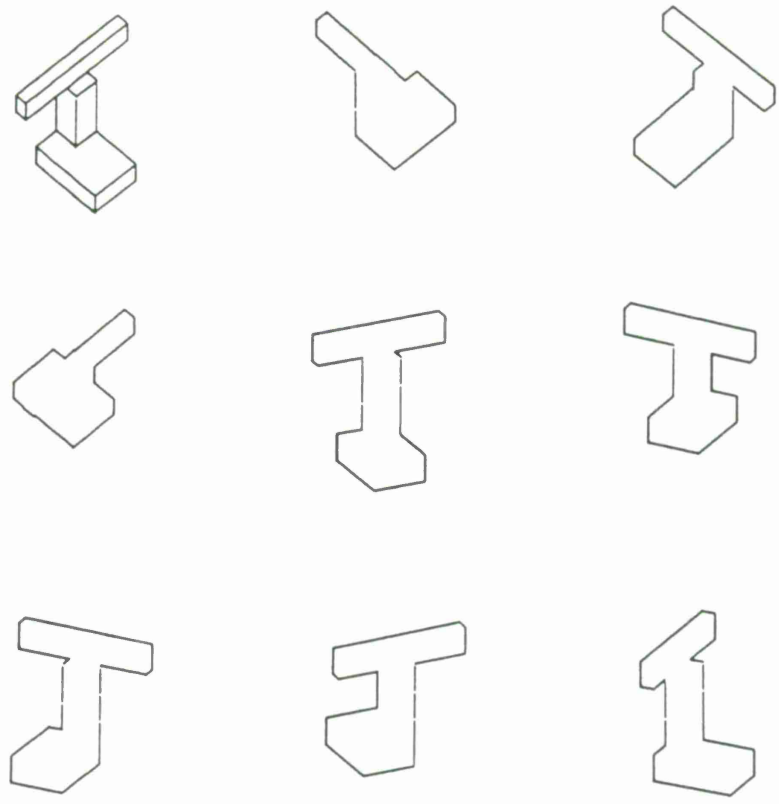
In the context of images such as Fig. 1-2, it is important for the system to perform in the presence of the degradations itemized above. Indeed, the targets of interest may always be partially occluded by other objects in the scene. Occluding objects at a different range may be separated from the target by their range values in this example, and more generally by other cues such as color, texture and motion. When the target cannot be separated from the background or from occluding objects, the system must be able to distinguish the target from other objects in its neighborhood. In addition to these artifacts due to the structure of the scene, the input images may be noisy and the early processing of these images may produce false, erroneous or missing features. It is important for the system performance to be robust in the presence of these degradations. Finally, a good recognition system must enable the user to define complex object models. In this context, the current implementation of SILC falls short of the laser radar image application because it involves objects with internal articulations which are not available in the current SILC system.

The performance of SILC in the presence of unknown viewpoint is illustrated in Fig. 1-5. The system successfully recognized all 8 silhouettes in the figure; these correspond to 8 different views of the model displayed in the upper-left corner of the figure. The characteristics of SILC are further discussed and illustrated in Section 3.

The proposed system is a good candidate for practical applications, especially those involving unknown orientations, potential occlusions and noise. In addition, the strategy implemented in this system should be applicable to several other signal interpretation tasks beyond the recognition of silhouettes.

1.3.2 Basic Strategy

The identification of an input silhouette is performed in SILC by first parsing the silhouette into a set of straight edges, then successively comparing the configuration of these edges with the edges of each model in the database. The silhouette configuration is compared to a model by conceptually structuring all possible interpretations of the silhouette edges into a tree, then pruning this tree by testing simple necessary constraints on the configuration of pairs of edges. The search efficiently discards most incorrect interpretation hypotheses; the remaining ones are



105340-5

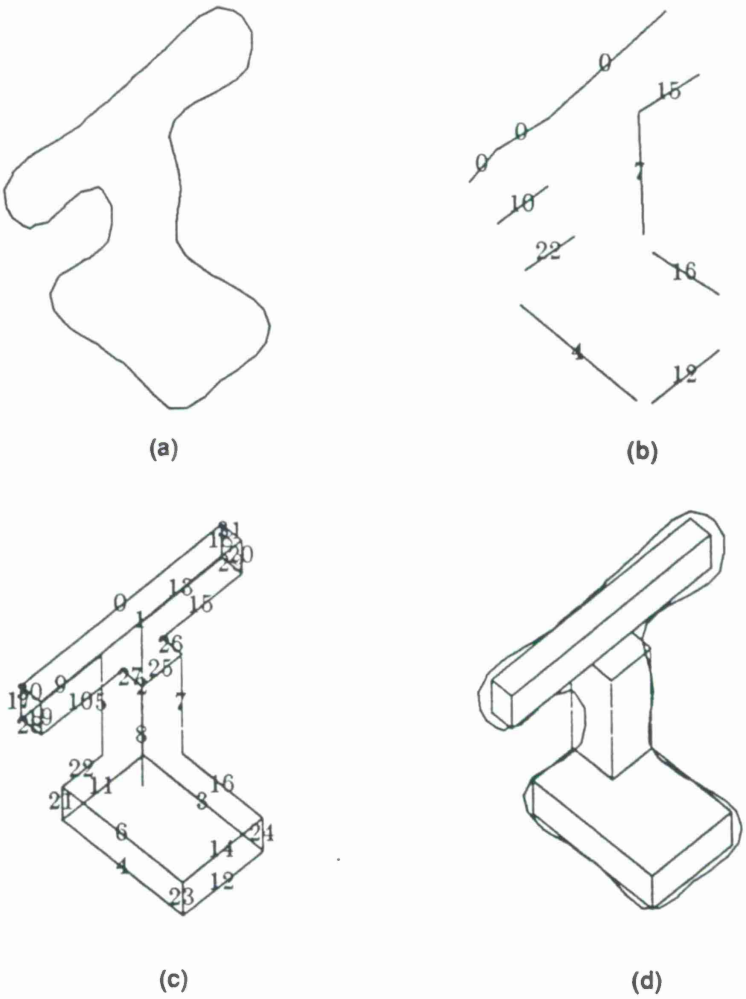
Figure 1-5. 3-D model and its silhouette for 8 views.

further tested by estimating the imaging transformation, synthesizing a silhouette of the model for this transformation, and finally comparing the synthetic silhouette to the observed silhouette. The various steps are illustrated in Fig. 1-6.

The silhouette in Fig. 1-6(a) is first processed and parsed into the set of straight edges pictured in Fig. 1-6(b). The tree search determines the valid associations of these edges with edges of the model shown in Fig. 1-6(c). Finally, an interpretation of the image edges such as that illustrated by the labels in Fig. 1-6(d) is confirmed by superimposing a synthetic silhouette of the model with the image edges.

1.4 ORGANIZATION OF THE REPORT

The remainder of this report is organized as follows. The second section of this report reviews some related work on model-based vision, both to set our system in the context of the current state-of-the-art, and to review some techniques that our system exploits. The third section outlines our strategy to the silhouette recognition problem, and shows how this strategy permits recognition in the presence of unknown view, occlusions and noise. The fourth section focuses on the key novelty in our search of the interpretation tree, namely the careful derivation of constraints on the relative positions in which a pair of edges can appear in the silhouette, given



105340-6

Figure 1-6. Example of recognition: (a)silhouette, (b)silhouette edges, (c) 3-D model, (d)recognized view.

their geometries in the 3-D model. The fifth section discusses our description of 3-D models and 2-D silhouettes. The compilation of tables predicting the model appearance in the image is discussed, as well as the evaluation of corresponding tables for the observed silhouette. The sixth section is devoted to the search of the interpretation tree; although the concept itself is well known, we indicate a number of heuristics used in SILC to improve the performance of the search. The verification of a candidate interpretation is discussed in Section 7, covering a number of novel approaches introduced in this work. In Section 8, the performance of the system is illustrated with a number of individual examples, and with performance statistics over moderate data samples. Finally, Section 9 concludes by suggesting some direct applications and directions in which the present work may be extended. A few questions of more technical interest are covered in the

appendices to this report, namely the parsing of silhouette chains and the estimation of a viewing transformation given correspondences between image edges and model edges.

2. BACKGROUND

In this section, a number of recently developed vision strategies applicable to silhouette identification are reviewed. Differences among these state-of-the-art systems and their influence on system performance are discussed. To conclude this section, a theory of silhouettes supporting the approach developed in the present work is reviewed.

2.1 BIBLIOGRAPHY REVIEW

Various strategies have been proposed for silhouette identification and several strategies are applicable to this problem. Four state-of-the-art systems are discussed in detail, namely the ACRONYM system developed by Brooks [5], the RAF system developed by Grimson and Lozano-Perez [11], the stochastic labeling designed by Bhanu and Faugeras [3], and the SCERPO system by Lowe [18]. Among these systems, only ACRONYM and SCERPO address the identification of 3-D objects in 2-D images taken from unknown viewpoints. The RAF system and the system by Bhanu and Faugeras only recognize 2-D objects in 2-D images. Note that Grimson and Lozano-Perez have also applied RAF to the recognition of 3-D objects given 3-D data; since we do not consider 3-D data, this system is not covered here.

All the above systems base their recognition on descriptions of images in terms of features. The choice of feature-based methods is first justified by comparing it to correlation-based methods. Then the choices of features and choices of hypothesis space search techniques will be addressed. Finally, the effects of these choices on system performance will be discussed.

2.1.1 Correlation-Based VS Feature-Based Recognition

Correlation-Based Recognition

One of the earlier silhouette identification approaches includes correlating a template of the object with the image and thresholding the resulting signal. Although successful fixed-font character readers have been built with this method, it has a large number of drawbacks in the context of more general tasks. Identification by correlations is the optimal algorithm in the statistical sense for recognizing templates in 2-D images with unknown translations and additive white Gaussian noise. However, it does not easily perform in the presence of unknown rotations and scaling in the image plane and cannot be used in the presence of unknown 3-D rotations, perspective and occlusions. The inability of correlations to address the more general vision problems can be attributed in part to its attempt to make an identification decision immediately based on raw image pixels. It is now widely recognized [19] that powerful and robust vision systems should be based on several levels of interpretations connecting the raw pixels in the input image to higher level decisions on the contents of the scene pictured in the image. This idea has been incorporated in the statistical pattern matching strategies by replacing correlations on regions by correlations on intermediate level descriptions such as chains of points on the region contour, or numbers

characterizing the global shape of the objects [8,14,23,30]. With some of these methods, it is possible to recognize 2-D templates with unknown orientation and scaling, but they all fail to address 3-D transformations.

The approaches based on classical pattern recognition techniques all match 2-D models with the image data. Images of 3-D objects taken from an unknown viewpoint can be identified with these methods only when the object is represented by a catalog of views. These catalogs are large for unrestricted orientations, to a degree where storage, computation and sometimes false alarm rates become impractical.

Feature-based Recognition

Large catalogs of views characterizing the appearance of a 3-D object in 2-D images can be avoided by performing the match directly between appropriate descriptions of the 2-D image and corresponding descriptions of the models. Image descriptions often consists of a set of features describing the image data by their nature and their configuration.

A major issue when matching descriptions in terms of features is that the identity, position and orientation of the object are initially unknown, and that the relations between 2-D features and 3-D features are also unknown at first. When the identity, position and orientation of the object can be hypothesized, correspondences are easily found and the hypothesis can be verified or invalidated. Similarly, when relations between image features and model features are hypothesized, the position and orientation of the object can easily be estimated and the relations can be tested. However, the hypothesis spaces for position and orientation on one side, and for feature pairings on the other side are generally much too large to be explored exhaustively. Techniques have been developed to efficiently search the space of viewpoints with the use of characteristic views [16,6,15], or a combination of the appearance models of Selfridge [22] and characteristic views [29]. Although good performance levels have been demonstrated with these methods, they suffer from a number of disadvantages. The performance of these algorithms is not guaranteed in the presence of occlusions and/or multiple objects; also, model compilation must often be assisted by the user for all but the simplest shapes. For the time being, the authors believe that matching in feature space is preferable for recognizing objects in the presence of occlusions and/or superimposed objects. The remainder of this review will therefore concentrate on the four model-based vision techniques mentioned earlier. These techniques basically explore the hypothesis space of matches between image features and model features, although some of them also estimate and exploit the viewpoint during the search.

2.1.2 Image Features

Descriptions of object images by features have been proposed in terms of corner points [21], straight edges [11], corners [24], parallel lines [18], generalized rectangles [5] or other features [4]. Generally, simple features such as lines or points each convey very little information about the position, orientation and identity of the corresponding object so that the recognition must be based more on the configuration of a set of features than on the characteristics of each single

feature. When operating with simple features, a substantial number of image features (say 5 to 10) must be matched to model features in order to claim a match with a reasonable confidence. The number of hypotheses is exponential in the number of image features [10], so that recognition based on simple features is confronted with an extremely large search space and is practical only with efficient search algorithms.

To avoid the search space explosion experienced with simple features, more complex image features can be considered, such as generalized rectangles, or images of circles. Complex features each retain more information on the identity and localization of the object so that an object can be recognized with a small number (1-3) of features. However, it is difficult to define sets of complex features that will accurately represent a wide variety of image shapes. In addition, their extraction from input images is performed by moderately complex open-loop image processing algorithms that are less robust than the algorithms for extracting simple features. ACRONYM is an example of a system based on relatively complex features, namely generalized ribbons. Good performance was demonstrated for the characterization of images of airplanes on the ground, but other types of scenes may be difficult to describe in terms of generalized ribbons. In addition to this issue, the image processing subsystem of ACRONYM was missing a substantial number of image ribbons in the example images, and it had difficulties detecting partially occluded ribbons.

2.1.3 Hypothesis Space Search

The hypothesis space of matches between image features and model features is now considered, and techniques for searching this space are discussed. The discussion is organized into a number of choices that were made in the four systems being reviewed

Search and Verification

In feature-based vision, the hypothesis space consists of all the possible pairings of image features with model features. When complex features are used, there are usually few possible pairings and the hypothesis space can be tested exhaustively; this is done for example in the ACRONYM system. However, when the image features are indistinguishable (for example, simple points), each image feature can be interpreted as any of the model features; for moderate numbers of image features, the size of the hypothesis space is astronomical in all practical cases so that an exhaustive search is excluded. To offset the huge size of the hypothesis space, the search usually proceeds in two phases. First, a small subset of the hypothesis space is selected using simple tests, then more comprehensive tests are applied to the remaining hypotheses. These two phases, referred to as search and verification, may be interleaved in practice.

Exhaustive or Satisfying Search

The search for candidate interpretations in RAF will provide all legitimate interpretations of image features (exhaustive search), while the search in the other systems will determine one of a few matches that satisfy the search constraints (satisfying search). After an exhaustive

search, the system evaluates confidence functions for all the retained hypotheses, and the search tests guarantee that all other interpretations will have lower confidences. The system can then select the optimal solution in the statistical sense by comparing the confidences of all selected interpretations. With a satisfying search, the system will find a few valid interpretations of the data with no guarantee of optimality. Although this approach finds a satisfactory explanation of the data, this may not be the only possible interpretation. For example, the silhouette of a symmetric object always corresponds to two symmetric views of the object, as illustrated in Fig. 2-1. Both views will be selected and will be given equal confidences with an exhaustive search; a satisfying search may find only one view, possibly the wrong one.

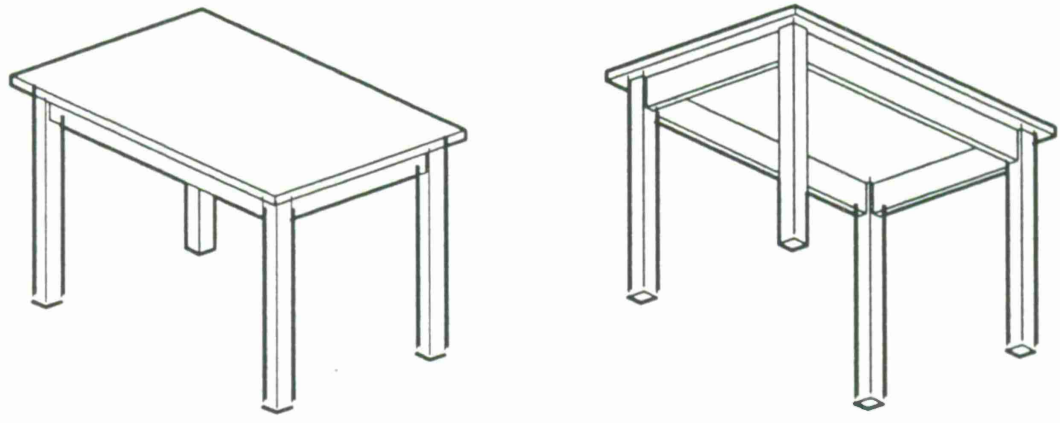


Figure 2-1. Two views of a symmetric object produce the same silhouette.

The suboptimality of a satisfying search may be an advantage in some cases, for example when the image features may be interpreted in terms of model features in a large number of similar ways. In such a case, the exhaustive method will spend large amounts of computation in testing each individual interpretation while a non-exhaustive method may investigate only a few interpretations in detail and find a satisfactory solution.

Search for a Complete or Sufficient Interpretation

The four systems also differ in the number of image features matched in the first stage of the search. The RAF tree-search (Grimson & Lozano-Perez) and the relaxation search (Bhanu & Faugeras) find an interpretation for all image features, whereas the grouping technique in SCERPO (Lowe) initially finds interpretations for a minimal set of features only. A system that interprets all the features in the first stage spends a large fraction of its computational efforts in the search for these interpretations, while the other system minimizes the search time by attempting to verify a hypothesized interpretation as soon as the number of matched features is sufficient to estimate a transformation between model space and image space. These systems will usually perform faster; however, the interpretation of a redundant set of features provides a better performance in the presence of image degradations.

Consistency Tests

Various tests can be applied to the data to select valid interpretations of features. In the systems reviewed here, the tests compare the geometry of image features with the geometry of the model to determine compatible interpretations of image features in terms of model features. A complete test of this compatibility includes verifying the existence of a transformation from model space to image space that will superimpose image features and model features. It is implemented by first estimating the transformation given the interpretations of image features in terms of model features, then by computing a synthetic image of the model for this transformation, and finally by comparing the image features with the features in the synthetic image. This test, referred to here as verification by synthesis, is applied during the verification stage in all the systems mentioned above. Because of the computational cost of estimating the transformation and of computing a synthetic image of the model, this test is not applied in the initial search stage. Instead, much simpler tests are applied to the data during the search phase; these tests are discussed in the next two paragraphs. Verification by synthesis can be used only when the number of matched features is sufficient to uniquely determine the transformation. When the transformation is determined, the superimposition of a synthetic image of the model to the image data may suggest interpretations of image features that were previously unmatched so that the labeling of image features can be extended and the transformation can be estimated more accurately. This extension of the match after verification is fully exploited in SCERPO, where only the minimum number of features are interpreted before attempting a verification. The minimum corresponds to the number of feature that will guarantee a unique imaging transformation. After this first verification, the interpretation is iteratively extended with the verification by synthesis.

Groupings of Image Features

In Lowe's SCERPO system, the initial feature interpretations are hypothesized by first detecting special viewpoint-independent configurations of edges in the image. Configurations of edges are selected for parallelism, colinearity and adjacency. Based on these configurations, the system determines candidate model edges to match these image edges. A complete database of objects can be precompiled to sort the configurations of model edges according to their predicted appearance in the image. It is then possible to find candidate matching model edges directly from this list. With this procedure, the system simultaneously tests for both the identity and localization of the object. The indexing into the entire database is important when matching a silhouette with a moderate to large database of objects. Very good results have been demonstrated with this method on real images of 3-D objects taken from unknown viewpoints, where the images include internal details of the objects. However, the system has not been demonstrated with silhouettes, and it can be argued that the grouping criteria perform well only when the models contain pairs of parallel edges and when these edges are visible in the image.

Constraints on Pairs of Edges

In the RAF system designed by Grimson and Lozano-Perez, and in the system by Bhanu and Faugeras, consistent interpretations are selected by comparing the configuration of each pair of image features (edges) to the configurations of pairs of model features. In the RAF system, strict bounds are set on the configurations of each pair of edges; when a pair of interpretations is determined to be incorrect, all the interpretations including the assignment for that pair can be rejected. This procedure is operated efficiently on a tree structure embedding the whole hypothesis space. In the system by Bhanu and Faugeras, confidence measures determine, at each moment, the belief that a given image edge can be interpreted as each one of the model edges. The degree of match between the configuration of each pair of image edges with each pair of model edges is used to iteratively update these confidence measures until the interpretation can be decided. Both techniques have the potential of incurring large computational costs since the problems they are addressing have an intrinsic exponential complexity. However, Grimson has shown, both theoretically and experimentally [10], that the RAF system will perform with very reasonable computational efforts in practical cases.

Decision of Inequalities in ACRONYM

In the ACRONYM system, the complex image primitives reduce the hypothesis space to a size that is easily searched exhaustively. Therefore, only "verification" tests are performed on the data. The problem addressed by ACRONYM is substantially more complex than that solved by the other systems, since the object models in ACRONYM allow for variations in internal parameters. For example, a model was designed for generic "wide-bodied passenger airplanes" in ACRONYM. This model has a range of acceptable values for body width, wing span, and for other parameters. The verification by synthesis is much more complex in this case, since it involves the decision of whether a system of equalities and inequalities has a solution or not. The decision of large sets of nonlinear inequalities is a very complex problem and its implementation is a key to the success of ACRONYM. In the implementation reported by Brooks [5], bounds are used to determine when a solution cannot exist. The implementation is quite successful at detecting airplanes on runways from a viewpoint close to the vertical. However, no examples are shown to support recognition from a horizontal viewpoint, such as that of a person standing on the runway.

It is interesting to contrast the ease of constraint tests on the configuration of each pair of edges in RAF with the difficulty associated with the simultaneous test of all constraints in ACRONYM. In both ACRONYM and RAF, the tests are guaranteed to retain all correct interpretations of the data. The strict necessary and sufficient conditions for accepting an interpretation are coupled and their equations nonlinear. In RAF, this complexity is avoided by using decoupled constraints on configurations of pairs of edges. These constraints are only necessary and not sufficient, thereby allowing some incorrect interpretations to be retained. In ACRONYM, the system corresponding to all strict constraints is built, but this system cannot be solved exactly because of its complexity. Brooks has proposed approximate solution methods for the constraints,

but these provide sufficient solutions so that incorrect interpretations may also be introduced here. The major difference between the tests in RAF and ACRONYM is then that in RAF, the constraints are decoupled using geometry in the problem domain, whereas in ACRONYM they are decoupled in their algebraic form. The method in RAF has the advantage that the decoupling is performed in advance so that tests can be compiled before accepting the input image. Precompilation of constraints would be much more difficult in ACRONYM.

2.1.4 Summary

A number of state-of-the-art model-based vision systems were reviewed in this section. These systems differ in several ways, by the features they extract from the images, by the method used for searching matches between image and model, by the emphasis on search or on verification, and by the tests applied to determine matches. By comparing the different approaches and their potentials, we decided to explore the application of tree-search techniques to the recognition of silhouettes, similar to those in RAF. The SILC system reported here is an implementation of this extension; in addition, it incorporates some of the iterative verification strategies developed by Lowe. The SILC system was also inspired by an understanding of silhouette theories developed by the author in other work [26]; a review of relevant aspects of silhouette theory are developed in the next section.

2.2 RELATIONSHIP BETWEEN 3-D OBJECTS AND THEIR SILHOUETTES

This section discusses the relation between the shape of a 3-D object and the shapes of its silhouettes. More specifically, we review the aspects of these relations which are relevant to the present work; a comprehensive analysis of the subject can be found elsewhere [26].

The word "silhouette" will be used to refer to outlines of images of objects in the projection plane. The silhouette of any object in an image is the projection onto the image plane of a set of special points on the image surface, which we will refer to as the silhouette generator. Points on the silhouette generator are the points of the object where "viewing rays" graze the object. For a smooth object, these points have a normal orientation perpendicular to the viewing rays (see Fig. 2-2). On a polyhedral object, the silhouette generator comprises the edges adjacent to one visible face and one hidden face (see Fig. 2-3).

An interesting question is to determine the set of viewpoints for which a given point on an object will appear on its silhouette. A point on a smooth surface is on the silhouette generator only for viewing directions parallel to the tangent plane at that point. However, a point on the edge of a polyhedron is on the silhouette generator when one face adjacent to the edge is visible while the other is hidden. In the case of orthographic projections, a face is potentially visible only if its normal has a positive component in the direction of view. In a representation of viewing directions by points on a unit sphere, the set of directions for which a face is visible is a hemisphere. The set of views for which an edge is on the silhouette is the double crescent delimited by the great circles parallel to the two adjacent faces, as illustrated in Fig. 2-4.

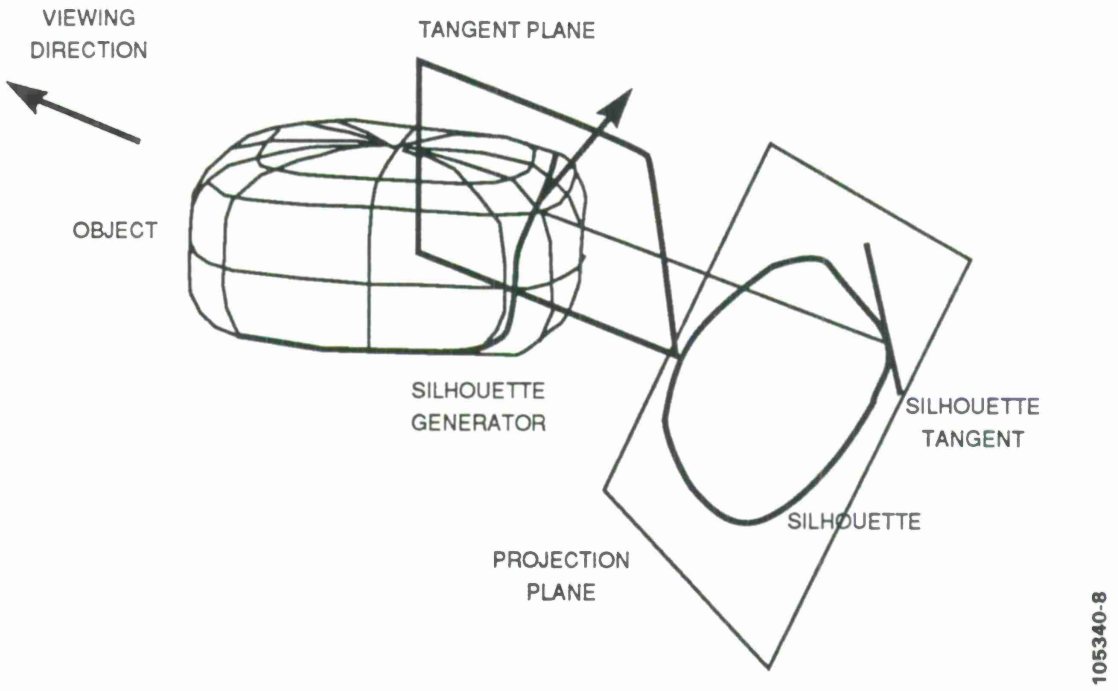
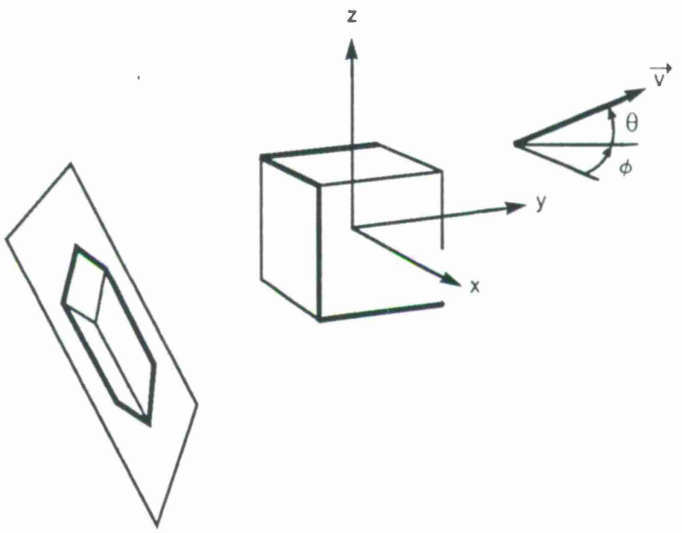


Figure 2-2. Silhouette generator for a smooth object.

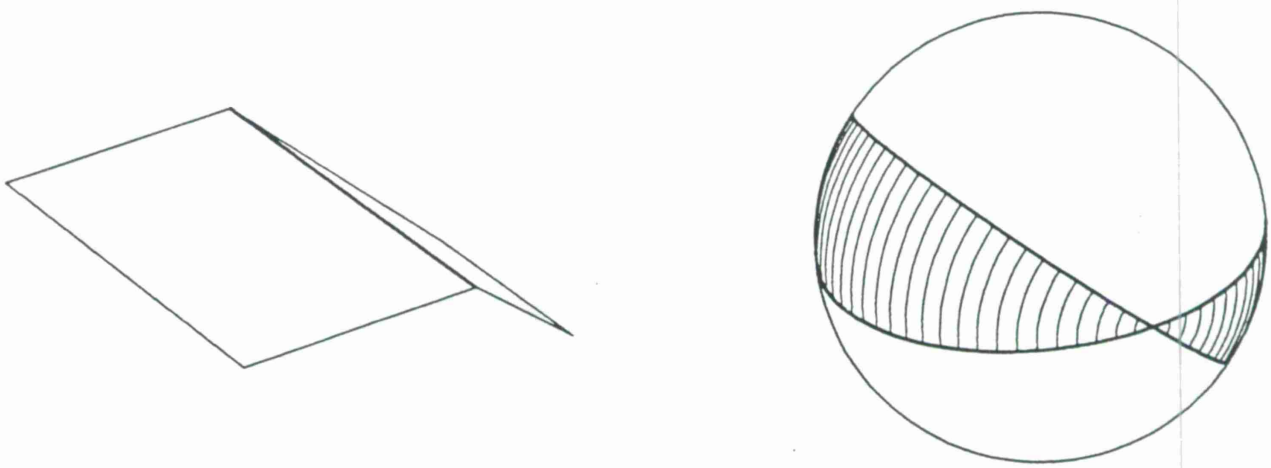
The above discussion was deliberately limited to convex objects, where an edge can be occluded only by the adjacent faces. An edge of a non-convex object is hidden for all viewpoints outside the double crescent corresponding to the edge, but it may not be visible for all viewpoints inside the crescent because of potential occlusions by other object parts. For a non-convex object then, the crescents are a superset of the viewpoints for which the edge appears on the silhouette. The set of views for which a model edge appears on the silhouette are crucial in the development of the appearance of a 3-D model in its silhouette for unknown viewpoints; this topic is developed in Sections 4 and 5.

The SILC system presented in this report was designed to recognize both polyhedral objects and objects with curved surfaces, given models of either the true object shape in the first case or a model of a polyhedral approximation to the object in the second case. It is interesting to note however, that there are substantial differences between polyhedra and curved surfaces in terms of the relations between silhouette shapes and object shapes. Given a point on the silhouette of a smooth object, the normal orientation of the surface at the corresponding point on the object silhouette generator is completely determined [2]. Therefore, the correspondence between a silhouette point and an object point determines strong constraints on the imaging transformation (4 constraints). For a smooth object, the silhouette generator varies continuously with changing viewing directions; any point on a smooth convex surface will be on the silhouette generator for some viewing direction. In the case of a polyhedral object, however, the silhouette generator can lie only on the convex edges of the polyhedron. In addition, the correspondence



105340-9

Figure 2-3. Silhouette generator for a polyhedral object.



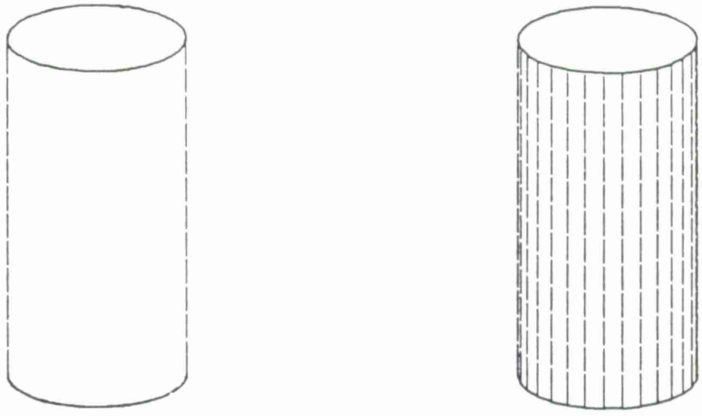
105340-10

Figure 2-4. Map of the viewing directions for which an edge appears on the silhouette.

between a silhouette edge point and a model edge point restricts the viewing direction only to a region such as the crescents in Fig. 2-4.

In the next few paragraphs, we will see how the differences discussed above between smooth

objects and polyhedrons can be reconciled in the context of our recognition system, and more specifically, we will determine how a polyhedral approximation to a curved surface can model the relation between this surface and its silhouettes. This discussion will be illustrated by the simple example of a straight cylinder approximated by a regular prism. Consider a straight circular cylinder and its approximation by a regular straight prism with 36 facets, as shown in Fig. 2-5.



105340-11

Figure 2-5. Cylinder and its approximation by a prism.

For any point on the lateral surface of the cylinder, there is a viewing direction for which the point will project onto the silhouette. When a match is hypothesized between a point on the silhouette and a point on the model, the viewing direction is restricted to be in the tangent plane at the model point, and the intersection of the tangent plane with the image plane must include the silhouette point. There is hence only one free parameter in the transformation, namely the angular elevation of the viewpoint with respect to the cylinder axis. The translation of these concepts in terms of the polyhedral approximation by a prism is now addressed.

The appearance of model points on the silhouette is considered first. Among points on the lateral surface of the prism, only those on the edges of the prism may be projected onto the silhouette for some viewing direction. In contrast, any lateral point of the smooth cylinder can appear on its silhouette. By making the approximation sufficiently fine, however, the sets of candidate silhouette generator points on the lateral surface can be made arbitrarily dense. A silhouette generator point can be made arbitrarily close to any given point by an appropriate choice of the approximation. In our example, a point on an edge of the prism approximation is at most 5 degrees away from any given point on the cylinder; this edge point will appear on the silhouette for appropriate viewpoints.

The constraints on the viewing direction determined by a match between a silhouette point and a model point are now considered both for a cylinder and for its approximation by a prism. When a lateral point of the cylinder is declared on the silhouette generator, the viewing direction is constrained to be parallel to the tangent plane at that point; representing viewpoints by points on the sphere, this corresponds to the meridian of the viewing sphere perpendicular to the normal at the object point. On the other hand, when a lateral edge of the prism is declared on the silhouette

generator, the viewing direction is constrained to be in the double crescent corresponding to the edge, as in 2-4. Although the viewing direction is constrained to a one-dimensional space in the case of the cylinder and to a two-dimensional set for the prism, the difference is reduced when these regions must be enlarged to take noise effects into account. In addition, the crescent for the prism is only 10 degrees wide in our example, and could be made thinner by choosing a prism with a larger number of facets (see Fig. 2-6).

In summary, although there are theoretical differences between smooth surfaces and their polyhedral approximations in the context of silhouette generation, the practical effects of these differences are only slight and can be reduced to any level by the choice of a fine approximation.

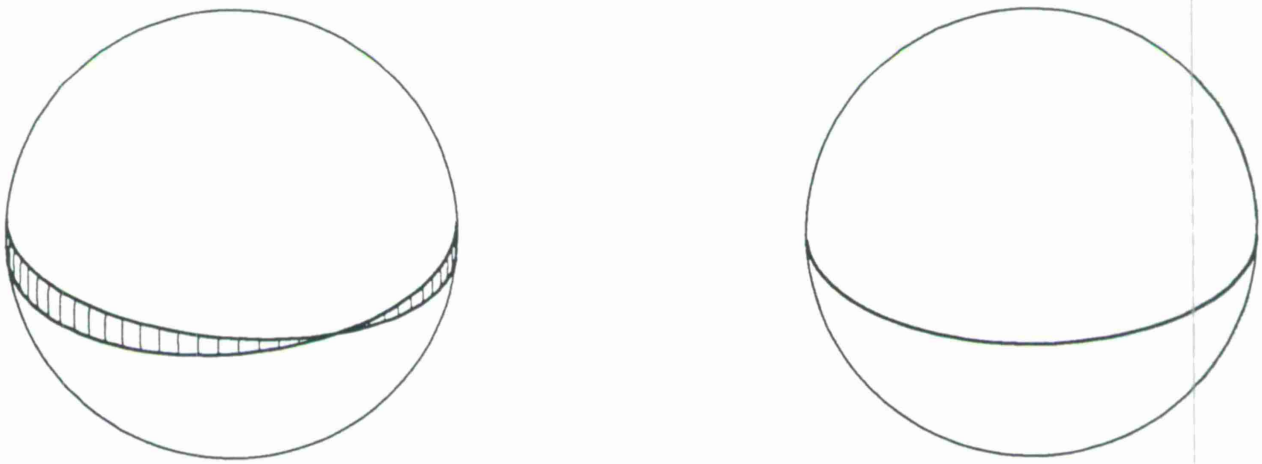


Figure 2-6. Viewing directions for a given silhouette generator.

3. SYSTEM DESIGN

In this section, the major specifications of the experimental silhouette recognition system (SILC) are detailed, and the system strategy chosen to simultaneously satisfy all performance criteria is developed and justified.

3.1 SYSTEM SPECIFICATIONS

SILC analyzes images of a scene which may contain one or more objects of interest. The system has a database of models for the objects of interest, and compares silhouettes in the image with the models in a database to determine if any of the silhouettes can be explained in terms of the models. When such an interpretation is discovered, it indicates a strong belief that the object in question is actually present in the scene. In the case where several interpretations are retained with a high confidence, the decision between them must be made by higher level processes. The system bases its decision on two types of data, the input images and object models; both are now discussed in more detail.

3.1.1 Input Images

The main input to the SILC system is an image of the scene. This image is expected to be a binary image of an orthographic project of the objects in the scene. Many times the binary image, which contitutes the silhouette to be recognized, is a result of some preprocessing stage that partially segments the object(s) of interest from the background in a gray-scale image.

The images provided to SILC may be acquired by a number of different sensors. In all cases, the images convey imperfect and/or incomplete information about the scene and the objects being imaged. The system must perform with incomplete knowledge of the imaging geometry and in the presence of degradations of the silhouette data. The major obstacles faced by the system are the ignorance of viewpoint, uncertainties on the scale of objects, the presence of occlusions and multiple objects in a silhouette, and the degradations in the image. These various facets are now discussed in more detail.

Unknown Viewpoint

Given a 2-D silhouette, SILC recognizes the 3-D object without any a-priori knowledge of the viewpoint. For example, all the silhouettes shown in Fig. 3-1 are correctly matched to the 3-D model of a table, illustrated in the upper-left of the figure. Note that the shapes of these silhouettes show wide variations over the various views; for example, the area, width, height, and elongation vary significantly from one view to the other.

In the current implementation of SILC, the projection is modeled as orthographic. This is a valid approximation of a general perspective projection for objects with a limited angular extent and imaged near the center of projection. In addition to the ignorance of viewpoint, it is assumed

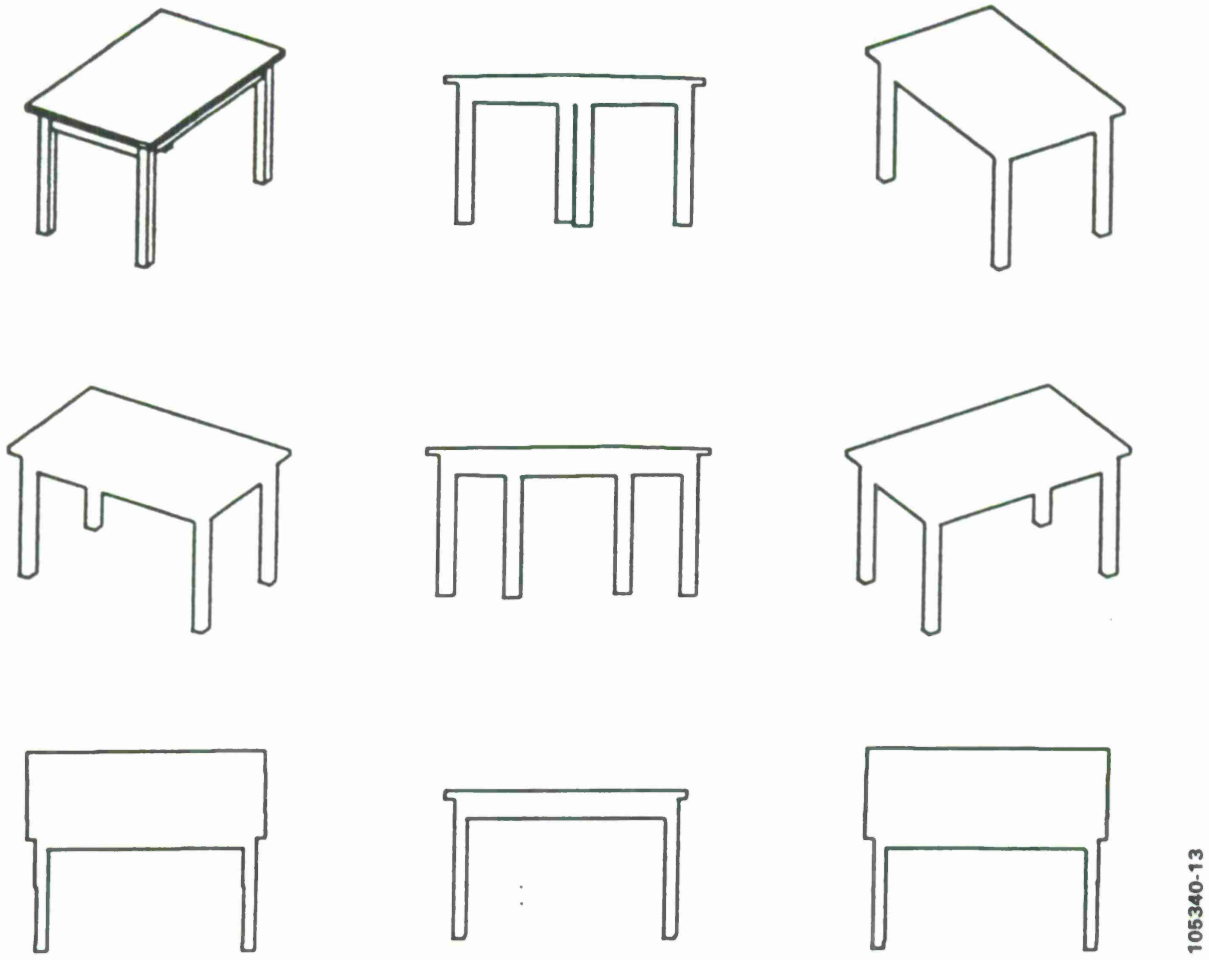


Figure 3-1. 3-D model and silhouettes for various viewpoints.

that the system has no reference to orientations and translations inside the image plane. The imaging transformation then has 5 degrees of freedom, namely three for rotations and 2 for translations. Note that the basic system concept is not restricted to orthographic projection; the extension to perspective projections should be straightforward.

Imprecise Scale

In a perfect orthographic projection, the scale of images in the projection plane is related to the scale of objects in the 3-D scene by the simple rule of foreshortening. However, when the orthographic projection is used to approximate a perspective projection, the relation between distances on the object and distances in its image also depends on the range of the object in the scene. The range resolution in laser radar images is sufficient to provide an accurate estimate of the image scale, but an accurate scale may not be available for other sensors. SILC successfully

recognizes objects when presented with silhouettes that are over- or undersized by 10 to 20%. For example, the table modeled by the polyhedron in the middle of Fig. 3-2 is successfully identified in the silhouettes to the left and to the right in the figure, which are scaled down and up by 15%. In favorable cases, the system may tolerate scale errors as large as 50% or more. Note that the orthographic transformation has 6 degrees of freedom when the scale is not precisely known.

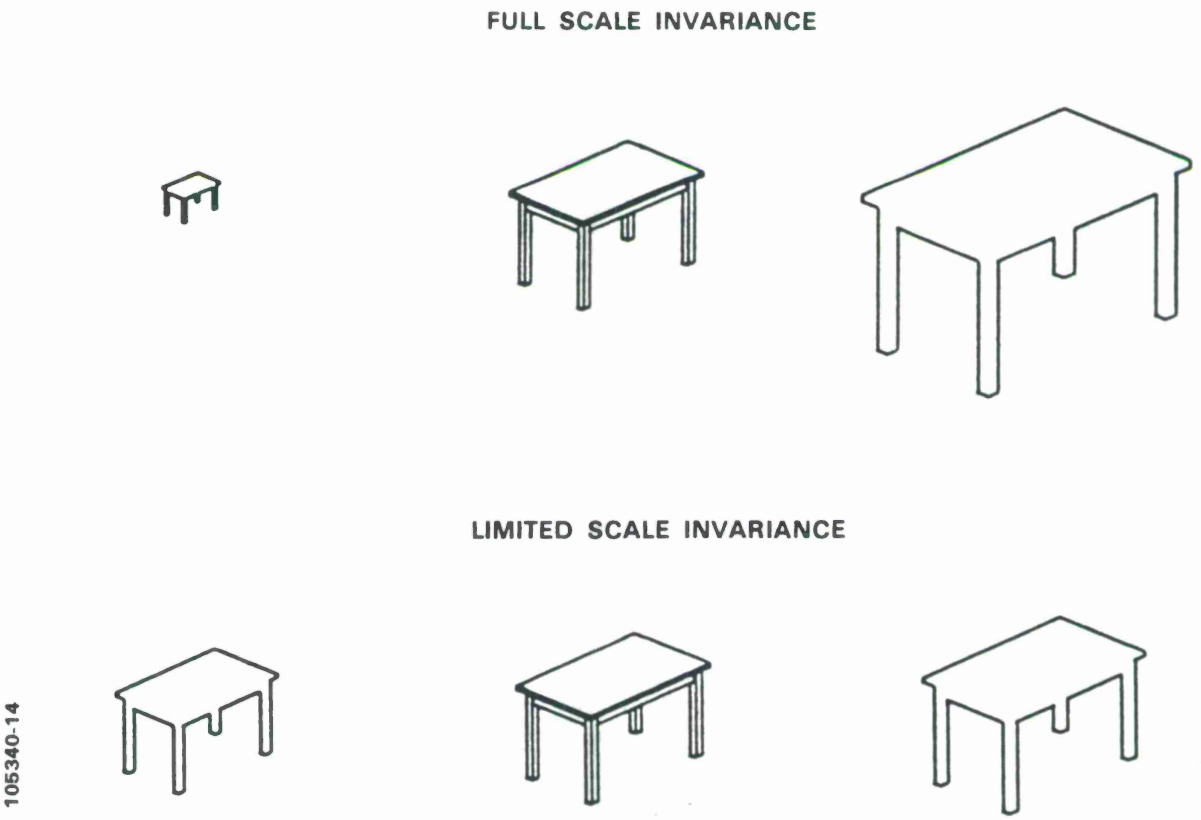


Figure 3-2. Scaled silhouettes recognized by the system.

Occlusions

Another important characteristic of SILC is that it will recognize objects in the presence of partial information. For example, when the object of interest is partially occluded by another object at a closer range, then the silhouette analyzed by the system matches only a part of the silhouette of the object model synthesized for the same view. In the example of Fig. 3-3, the table is partially occluded by the person standing in front of it; the figure also shows that the silhouette segments analyzed by the system cover only a part of the true object silhouette. As a particular case of occlusions, the system correctly handles self-occlusions in non-convex objects.

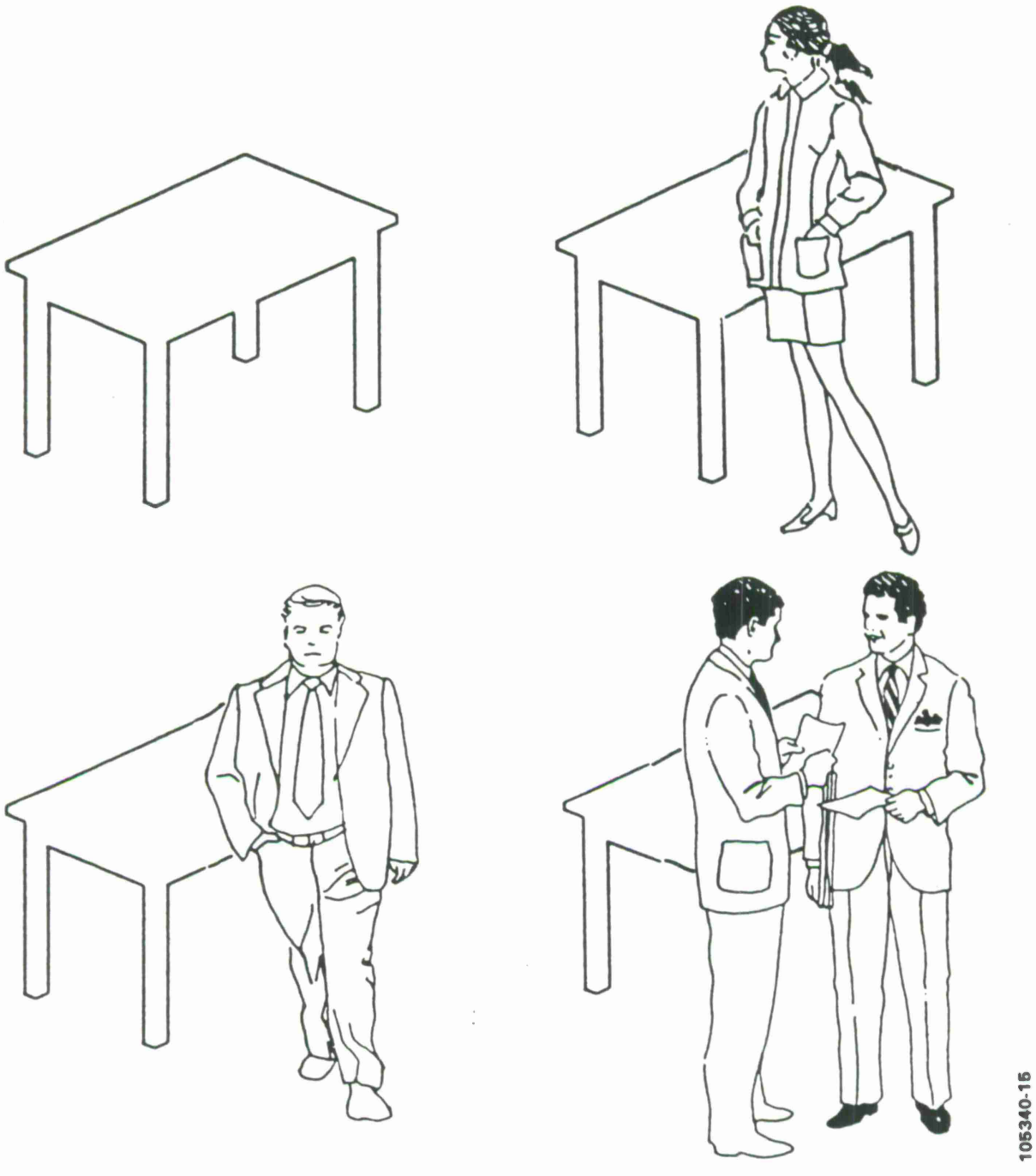
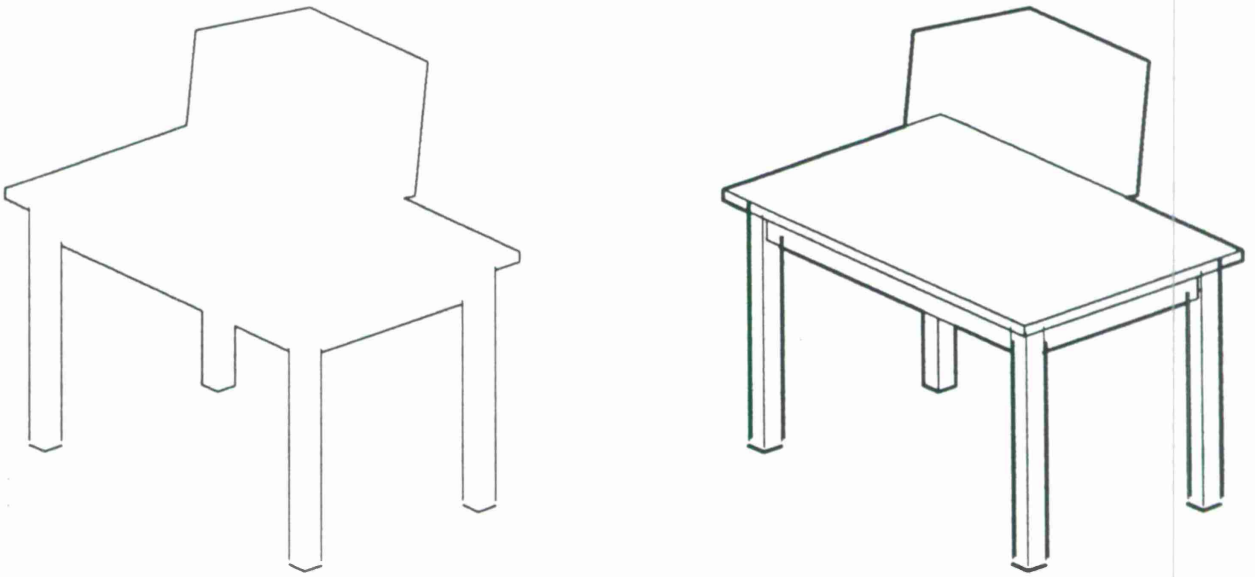


Figure 3-3. Example of partial occlusion.

Multiple Objects

In the above example, it was assumed that the occluding object can be separated from the object of interest in the image. This separation can be based on range gating in low-resolution

range images; it could be based on texture, shading, color or motion in passive optical images. However, it is not always possible to separate the object of interest from "clutter" in the scene. The system has the ability to recognize an object in a silhouette composed in part by the object of interest, and in part by other objects. In Fig. 3-4, the scene on the left contains the object of interest, the table, and another object, the computer terminal. The silhouette of this combination of objects, shown on the right, is composed in part by silhouette segments of the table, and in part by other segments. To achieve a successful recognition, SILC has to distinguish the silhouette parts corresponding to the object of interest from the other parts.



105340-16

Figure 3-4. Example of silhouette of multiple objects; left: scene, right: silhouette.

Image Degradations

In the design of a signal interpretation system, it is crucial to account for the presence of degradations in the input data. In particular, SILC has been designed to account for the presence of noise in the input images. The silhouettes presented to the system could be extracted from images generated by a number of different sensors. These include passive optical or infrared images, laser radar images and range/doppler images. Since each sensor may produce different noise characteristics, the system does not attempt to account for an accurate model of sensor noise. Instead, degradations are accounted for by allowing uncertainties in the silhouette shapes extracted from the image. In addition to degradations in the input images, other errors may be introduced by the early processing of the image data. These errors may consist of false or missing image features, and of errors in the estimates of feature characteristics. The identification system is designed to accommodate these errors by rejecting spurious features and by matching models with incomplete data.

Limitations

In the current implementation, the system processes a silhouette corresponding to a single object or perhaps to a handful of superimposed objects, and interprets this silhouette in terms of the object models stored in its database. When recognition is applied to a cluttered scene where the object of interest accounts for a small fraction of the edges, the system must allow an extremely large number of null edges and its performance becomes useless; the SILC system was not designed to directly analyze complex scenes as a whole. In order to perform scene analysis with a good level of performance, the present system must be coupled with an image segmenter and perhaps a top-level engine for controlling the focus of attention. See Section 3.2.2 for further discussion on the use of null edges in matching.

3.1.2 Object Models

The model database is composed of objects defined by rigid polyhedra. Note that this choice does not necessarily limit the system to the "blocks world", since complex shapes can always be modeled arbitrarily closely by polyhedra. The SILC system must be given a description of each model with sufficient detail to uniquely specify the geometry of a polyhedron. The description may be given in terms of faces, edges and vertices, or in terms of set operations on primitive volumes. It is assumed in the design of the system that the object models are given to the system ahead of time so that it can compile special representations to improve its performance while running the recognition algorithm. The specific form of the models and the process by which they are compiled is discussed in detail in Section 5.

3.1.3 Review Of Specifications

The main specifications of our silhouette recognition system discussed in the above sections are summarized below.

The system

- Recognizes 3-D Objects in 2-D Images
- Uses Only Silhouette Data
- Uses a Database of Polyhedral Object Models
- Performs in the Presence of
    - Unknown Viewpoint
    - Moderate Scale uncertainties (20%)
    - Occlusions, resulting in missing features
    - Superimposed objects, resulting in imperfect segmentation

- Image Noise, resulting in silhouette shape degradations
- Early vision artifacts, resulting in spurious/degraded features

3.2 SYSTEM STRATEGY

In this section, the recognition strategy applied by SILC is presented, and justified with respect to the specifications described above. The SILC system interprets a silhouette by comparing it successively to each model in the database. In the remainder of the report, only the problem of comparing a silhouette to one model is considered.

The match between the silhouette and a model is performed at the level of edge features. Since the identity of the silhouette is mainly retained in the edge configurations, the match consists of comparing edge configurations in the image to those in the model. The silhouette shape is uniquely described by a set of edge features only if their number is sufficient to prevent the image configuration from matching the model by pure chance. We have observed experimentally that six to ten edges are required in general to uniquely characterize the shape of a silhouette. In the absence of constraints, the number of interpretations of I image features in terms of M model features is M^I. Typical numbers of interpretations are on the order of 10^{20}, which is impractically large for direct evaluation. We have adopted a tree-pruning strategy similar to the one proposed by Grimson and Lozano-Perez to reduce the size of the hypothesis space. We test intermediate nodes of the tree by comparing the configurations of pairs of silhouette edges with the configurations of the matched pairs of model edges. We have carefully designed these binary constraints to make them viewpoint independent and to allow for the effects of scaling, occlusion and noise. The configuration of a pair of silhouette edges is tested by comparing six numbers describing their relative position to the ranges of these numbers predicted for the matched model edges. The numbers describing silhouette edge configurations are computed for all pairs before starting the tree search; the numbers for the model can be compiled off-line. The test of each tree node costs then only a few arithmetic comparisons and is extremely efficient.

Although the decoupled constraints on the configurations on pairs of edges are extremely simple, they are only necessary constraints on the validity of the match. As a consequence, the tree search is guaranteed to retain all correct interpretations of the data, but it may also retain some incorrect interpretations. It is therefore necessary to apply a final test to each interpretation retained by the tree search. The test is implemented in SILC by comparing the image data with an image of the model synthesized from an appropriate viewpoint. In addition to the acceptance decision, this comparison provides a confidence factor indicating the quality of the match.

The major operations of the system during recognition are

- Extraction of Image Edges
- Tree-Pruning of the Hypothesis Space of Edge Pairings
- Verification of the Retained Hypotheses

This overall system strategy closely follows the one described in [11]. However, substantial differences between the two systems can be found in the constraints used in the tree search, in the verification strategy, and in the compilation of model constraint tables. We will now discuss in more detail how the system strategy supports the system specifications described earlier, namely recognition in the presence of image degradations and the matching of curved silhouettes with polyhedral models. We first justify the exhaustive tree-search technique in the context of statistical pattern recognition, then show how the match of silhouette edges with model edges accommodates the specifications for the system.

3.2.1 Matching with Uncertainties

In this section, we discuss how the concept of pruning a large space of hypothesized pairings between image and model features is justified by concepts of statistical classification.

In the vast majority of signal interpretation tasks, the input signals are degraded by uncontrollable events, so that no signal ever matches a model exactly. Classical pattern classification approaches in the presence of degradations address this problem by decision methods such as maximum likelihood and maximum a-posteriori probability decisions. These consist of estimating the likelihood or posterior probability of each hypothesized event and choosing the event with the highest figure. In the presence of huge discrete hypothesis spaces, it is impractical to compute likelihoods or posterior probabilities for all events; the classification can be performed only if a majority of hypotheses corresponding to extremely low probabilities can first be discarded. In the classical analysis, hypotheses can be discarded only when the probabilities associated with these are zero; in turn, null likelihoods or posterior probabilities will be experienced only when the noise model for degradations proposes zero prior probabilities for degradations outside some finite interval. The decision system can then reject a hypothesis categorically if its probability or likelihood can be proven to be exactly zero.

The SILC system must be given strict bounds on all the degradations and free parameters (except for the viewpoint); the prior probability densities are set to zero outside those bounds. The signal interpretation task can then be separated in two parts. The first part consists of discarding all the hypotheses corresponding to zero likelihoods or posterior probabilities. The second part consists of estimating a figure of merit for each remaining hypothesis and of selecting the best interpretation based on this figure. In this report, we focus the attention on the first part of the signal interpretation. The goal of our system is hence to determine all legitimate interpretations of the data, given bounds on the amplitude of degradations and on the values of free parameters. The system will always select the correct interpretation of the data when there is one. In addition, the system will also select interpretations that are consistent with the input data and the given bounds on degradations, even when these interpretations do not correspond to the real configuration of the scene. In particular, all symmetric orientations of a symmetric object are selected by the system. When the system is given loose bounds on image degradations and on free parameters, the system is likely to retain additional interpretations of the data. These additional solutions can be considered as "false positives" in the statistical sense, but they are inherent in the data (signal + noise limits) presented to the system.

In summary, the system accepts noisy input images and bounds on the degradations of the image. In return, the system provides all interpretations of the data that are consistent with the known models and with the noise bounds. By design, the system has a false reject rate of zero, and its false alarm rate reflects the ambiguity of the input data.

3.2.2 Matching Edges

In this section, we discuss how the SILC system appropriately responds to the degradations listed in Section 3.1.3 by the tree-pruning search based on pairwise constraints and by the verification of the retained hypotheses. The compatibility of pairwise constraints with degradations is further developed in Section 4, whereas the relation between degradations and the verification is investigated in Section 7.

Degradations of the positions and orientations of silhouette edges by noise are easily accounted for by relaxing the thresholds tested against edge pair configurations. This increase in the accepted range of image measurements is carefully controlled in SILC and tailored to the expected noise margins for each individual measurement. Occlusions in the scene must be modeled in a slightly different way. As a result of occlusions, certain edges may be only partially visible in the image or may even be totally absent. In the system, partially occluded edges are taken into account by always allowing the match of a partial image edge to a full model edge. This feature is also important in the matching of silhouette curves with polyhedral models. The total absence of an edge from the image is not an issue in the implementation since the system finds interpretations of silhouette edges in terms of model edges and doesn't require each model edge to be matched.

The case of multiple objects in the same silhouette is taken into account in the system by an additional "model" edge, namely the "null" edge. Silhouette edges corresponding to a different object in the scene are assigned to the null edge. The null edge may also be used to discard from the match, a spurious edge arising from an error in the early processing of the image data.

We stated earlier that because of occlusions, a match must be possible between an image edge and a model edge when the image edge covers only part of the model edge. However, the opposite does not apply, i.e., a model edge may not be matched to a longer silhouette edge. In the case of special alignments, the image may contain a long edge in the silhouette made of the alignment of two or more edges in the object. The system will not be able to correctly match this edge; if the silhouette contains a sufficient number of edges, the other edges may be matched and the interpretation will be accepted by interpreting the merged edge as a "null edge."

A key to the success of our implementation is the search for binary constraints which maximize discriminating power while always accepting correct matches, even in the presence of noise or other artifacts in known amounts. Powerful yet satisfying constraints are attained by a careful strategy for extracting long straight edges from the silhouette, an innovative limitation of the range of viewing angles considered for each pair of model edges, and by devising realistic upper bounds on noise degradations. The constraints on pairs of edges are developed in the next section.

Any recognition system must accept matches in the presence of foreseeable image degradations; a successful system must hence perform with imperfect input features. Our system considers two

types of degradations and responds to them in two different ways. Small deviations between model and observations are accounted for by carefully relaxing the constraints according to estimates of the image degradations. On the other hand, large degradations such as missing features, extra features and misinterpreted features are either implicitly covered by the system approach or treated as outliers and discarded.

3.2.3 Review of the System Response to Degradations

In the previous section, we discussed how degradations in the inputs can be addressed in the context of the edge matching strategy. In summary, the degradations are overcome by

- Relaxing the Constraints (Noise)
- Matching Silhouette Edges to Parts of Model Edges (Occlusions)
- Rejecting Spurious Edges (Multiple Objects)

3.3 SUMMARY

In this section, we have presented the main strategy used in the silhouette recognition system. The system is based on the matching of silhouette edges to edges of models in the database. The matching is performed by a tree-pruning search followed by a verification by synthesis. This approach can be tailored to perform in the presence of realistic image degradations and uncertainties about the imaging geometry.

In the next few sections of this report, we will present the implementation of each system part in more detail. Specifically, we will describe our implementation of the binary edge constraints in Section 4, the description and compilation of models in Section 5, detail relevant to the tree-search in Section 6 and the verification of candidate hypotheses in Section 7. In the remaining sections, we will analyze the performance of the system, both from an experimental and from a theoretical viewpoint, and present our conclusions.

4. CONSTRAINTS ON THE CONFIGURATION OF PAIRS OF EDGES

In this section, we discuss the constraints tested on each pair of edges in the interpretation corresponding to an intermediate tree node. These constraints ensure that the configuration of the pairs of edges extracted from the observed silhouette is consistent with the image configuration predicted for the matched model edges. In the absence of image noise and occlusions, and with a perfect knowledge of the viewpoint and scale of the image, the configuration of two silhouette edges must be identical to the configuration of two model edges to claim a match. However, in the face of noise, occlusions, and variations of viewpoint and scale, two model edges may appear in a range of different configurations in the image. The tests designed in this section verify only necessary constraints on the interpretation of a pair of silhouette edges in terms of a pair of model edges. In other words, the match is accepted if the configuration of the silhouette edges is consistent with the configuration of the model edges. The match is rejected only if the configuration can be proven to be incompatible with any acceptable values for viewpoint, scale, noise, and occlusion.

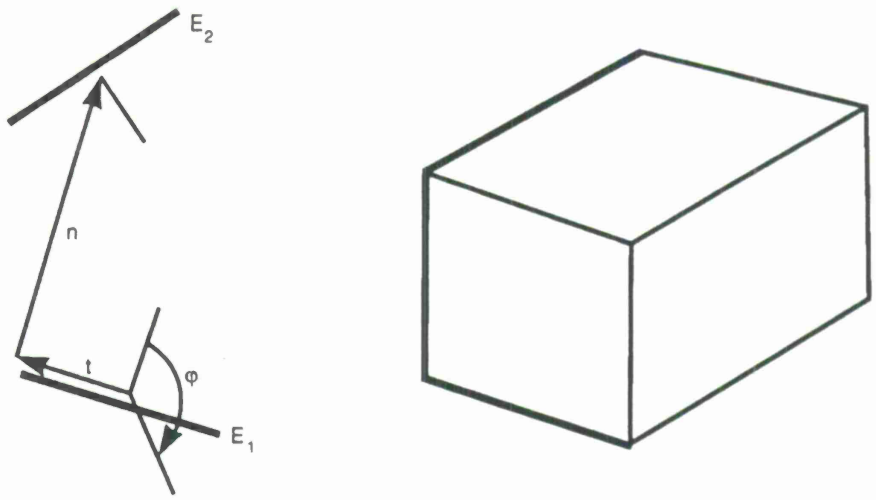
The constraint tests will be developed first for the simplest case, which has no image noise or occlusions and where the image scale and viewpoint are perfectly known. Then, the tests will be extended successively to include the effects of noise, occlusions, scale variations, and finally viewpoint variations.

4.1 CONSTRAINTS IN THE SIMPLIFIED CASE

In the absence of noise, occlusions, and variations of scale and viewpoint, the interpretation of a pair of silhouette edges as a corresponding pair of model edges can be accepted only if the configurations of the silhouette pair is identical to the configuration of the projection of the model pair, i.e., if these two pairs can be superimposed by a simple rotation and translation in the image plane (see Fig. 4-1).

To design the test, it is sufficient to completely characterize the relative position of the two edges, and to require that these characterizations be identical for the model edges and the silhouette edges. We have chosen to characterize the relative position of two edges by three measures, namely the relative angle ϕ between the edge normals, and the vector distance between the edge centers, where the two components of the distance vector are the component t (tangent distance) along the first edge and the component n (normal distance) along its normal.

It is assumed in this work that the direction of the outside normal of the edge can be extracted from the image, i.e., the early vision subsystem extracting the silhouette edges can also determine which side of each edge is inside the silhouette and which side is outside. With this assumption, the distance components n and t have a consistent sign. We have chosen the positive n axis to point outward from the silhouette, and the positive t axis to point in the counterclockwise direction when following the silhouette. The constraints on the interpretation of a pair of edges are tests on the equality between the signed values of t, n, and ϕ



105340-17

Figure 4-1. Configuration test in the absence of noise.

$$\begin{cases} t^C_{sil} = t^C_{mod} \\ n^C_{sil} = n^C_{mod} \\ \phi_{sil} = \phi_{mod} \end{cases} \tag{4.1}$$

where the superscript C indicate that the distances correspond to the center points.

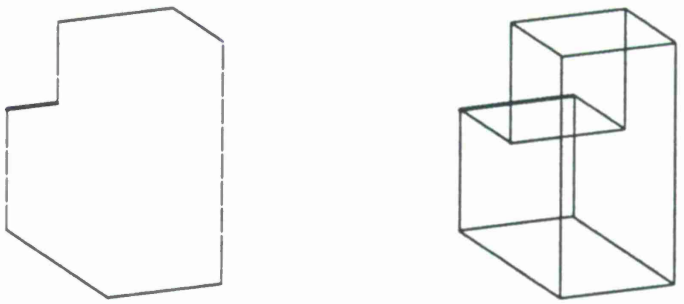
The choice of measures to characterize configurations of pairs is not unique, but the choice described above is well suited to the design of constraints in the presence of image degradations. This design also accommodates within the same framework both finite edges and zero-length edges used to characterize curves.

4.2 CONSTRAINTS IN THE PRESENCE OF OCCLUSIONS

In the presence of partial occlusions, either by the object itself for non-convex objects, or by other objects, edges extracted from the silhouette in the image may correspond to only a fraction of the corresponding edge in the model (see Fig. 4-2).

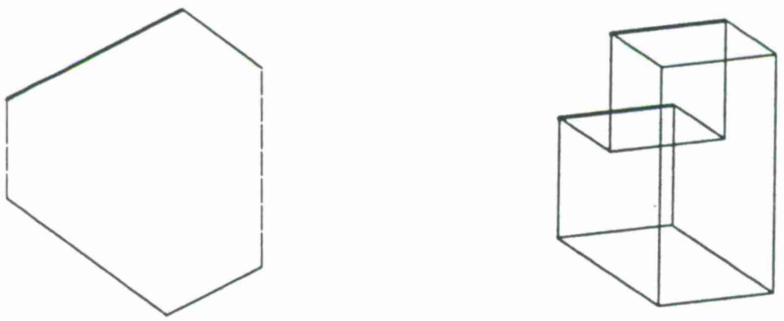
In addition, silhouette edges can be split because of image degradations, and zero-length edges used to represent curve points will match only a single point of the edge approximating the curve in the model. However, we do not consider the case where a model edge matches only part of an edge extracted from the silhouette; this case is rare but occurs in the presence of special alignments such as shown in Fig. 4-3. The system handles this case by leaving the combined edge as uninterpreted. In brief, the match of a silhouette edge and a model edge is acceptable when the silhouette covers part of the projection of the model edge, but not vice-versa.

Although the relative angle ϕ of two silhouette edges is independent of their length, the relative distances t and n may be different for different fractions of the projected model edges, since the center points of the edge parts may take different positions. The configuration test must determine



105340-18

Figure 4-2. Silhouette edge matching a fraction of the model edge (occlusion).



105340-19

Figure 4-3. Model edge matching fraction of silhouette edge (special alignment).

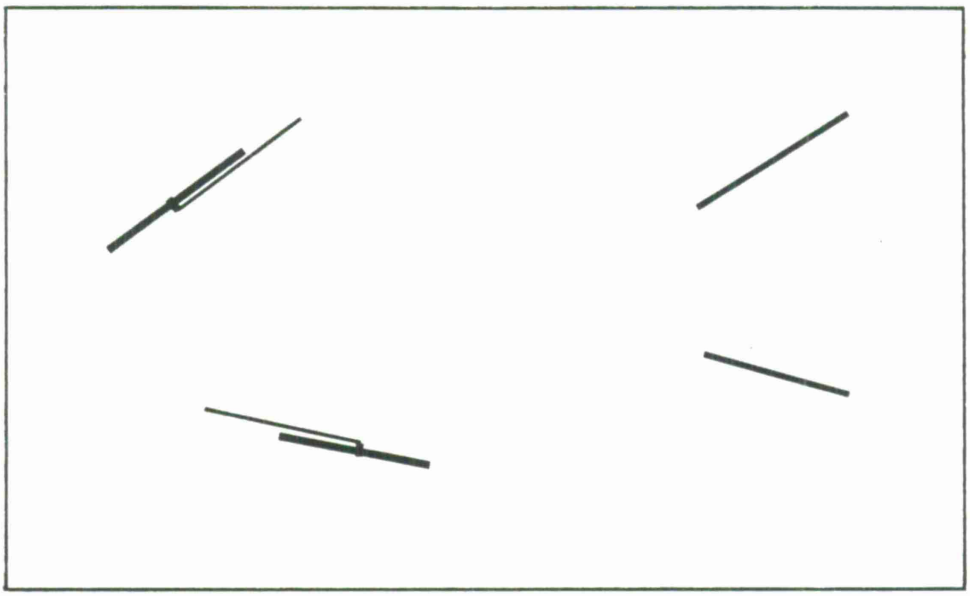
if the distances t and n are consistent with positions of the silhouette edges that fall within the projections of the corresponding model edges. The center point of each edge fraction in the silhouette may lie anywhere between the extreme points in the projected model edge. The test for distances between the midpoints is then

$$\begin{bmatrix} t^C_{sil} \in [t^C_{mod} - l^1_{mod}/2 - |\cos\phi| l^2_{mod}/2,\ t^C_{mod} + l^1_{mod}/2 + |\cos\phi| l^2_{mod}/2] \\ n^C_{sil} \in [n^C_{mod} - |\sin\phi| l^2_{mod}/2,\ n^C_{mod} + |\sin\phi| l^2_{mod}/2] \end{bmatrix} \tag{4.2}$$

where l^i_{mod} is the length of model edge i.

Although this test accepts all the valid configurations, it also accepts a large number of invalid configurations; for example, it will accept a configuration where both center points of the silhouette edges match the extreme points of the projected model edges, even though the finite length of the silhouette edges puts its endpoints out of range (see Fig. 4-4).

The drawback of the above test is that it is designed for silhouette edges of any length, including zero so that it doesn't take advantage of the known length of the silhouette edges. The longer the observed silhouette edges are, the stronger the constraints on the relative configurations on a pair of edges. In particular, when the lengths of the silhouette edges are equal to the lengths of the model edges, the tests in Eqn. 4.1 are applicable. To offset this unfavorable behavior of the test in Eqn. 4.2, the midpoint constraint will be replaced by a constraint on the ranges of distances between points on each edge. This constraint does exploit the estimated length of the observed



105340-20

Figure 4-4. Incorrect match accepted by the midpoint distance constraints.

silhouette edges to maximize the capability of rejecting incorrect pairings. It is equivalent to Eqn. 4.2 in the case of zero-length edges and to Eqn. 4.1 in the case where the lengths of the silhouette edges are equal to the lengths of the projected model edges.

The alternate test considers the distances n and t measured between all pairs of points on the two edges; the sets N_{sil} and T_{sil} contain all the values of t_{sil}, n_{sil} for the pair of silhouette edges. The silhouette configuration is accepted if these sets are included in the sets N_{mod} and T_{mod} representing the possible values of these distances between points on the projected model edges. The sets N_{sil}, T_{sil}, N_{mod}, and T_{mod} are all single intervals so that the inclusion tests are equivalent to a test of their limits.

$$\begin{cases} Max\ t_{sil} < Max\ t_{mod} \\ Min\ t_{sil} > Min\ t_{mod} \\ Max\ n_{sil} < Max\ n_{mod} \\ Min\ n_{sil} > Min\ n_{mod} \end{cases} \tag{4.3}$$

The minima and maxima on the right side of the above equations can be evaluated in terms of the angle ϕ_{mod} between the projected model edges, the distances between the midpoints of the edges, t^C_{mod}, n^C_{mod}, ϕ, and the lengths l^i_{mod}, l^i_{sil} of the model and silhouette edges. Note that these extrema are always attained when the point on each edge is at one of the extremities of the edge.

$$\begin{cases} Min\ t_{mod} = t^C_{mod} - l^1_{mod}/2 - |\cos\phi_{mod}|l^2_{mod}/2 \\ Max\ t_{mod} = t^C_{mod} + l^1_{mod}/2 + |\cos\phi_{mod}|l^2_{mod}/2 \\ Min\ n_{mod} = n^C_{mod} - |\sin\phi_{mod}|l^2_{mod}/2 \\ Max\ n_{mod} = n^C_{mod} + |\sin\phi_{mod}|l^2_{mod}/2 \end{cases} \tag{4.4}$$

Similarly, the extrema on the left side of Eqn. 4.3 can be evaluated in terms of the angle between the silhouette edges, ϕ_{sil}, the distances between the midpoints of the edges, t^C_{sil}, n^C_{sil}, and the lengths of the model and silhouette edges.

$$\begin{cases} Min\ t_{sil} = t^C_{sil} - l^1_{sil}/2 - |\cos\phi_{sil}|l^2_{sil}/2 \\ Max\ t_{sil} = t^C_{sil} + l^1_{sil}/2 + |\cos\phi_{sil}|l^2_{sil}/2 \\ Min\ n_{sil} = n^C_{sil} - |\sin\phi_{sil}|l^2_{sil}/2 \\ Max\ n_{sil} = n^C_{sil} + |\sin\phi_{sil}|l^2_{sil}/2 \end{cases} \tag{4.5}$$

4.3 CONSTRAINTS IN THE PRESENCE OF NOISE

In this section, the effect of noise on the configuration test for the interpretation of a pair of silhouette edges in terms of a pair of model edges is addressed. Before addressing that particular issue, however, we will analyze the more general problem of testing an estimate against bounds, when the estimate is corrupted by noise. We will then introduce a "noise model" for the edge measurements and finally derive configuration tests for pairs of edges in the presence of noise.

4.3.1 Testing Constraints on Noisy Estimates

We analyze the problem of testing whether the value of a variable v, for which we have an estimate $\hat{v}$, is contained within the bounds v_{min}, v_{max}. When the estimate is perfect, i.e., unbiased and with a zero variance, the test is simply

$$\begin{cases} \hat{v} > v_{min} \\ \hat{v} < v_{max} \end{cases} \tag{4.6}$$

The regions corresponding to the acceptance and rejection decisions are sketched in Fig. 4-5.

105340-21

Figure 4-5. Decision regions, noiseless estimate.

Consider now the case where the estimate $\hat{v}$ is corrupted by additive noise, and where the magnitude of this noise is bounded by N_v. In that case, given a value for the estimate $\hat{v}$, the true value of the variable v may be anywhere in the interval $[\hat{v} - N_v, \hat{v} + N_v]$. The test of the value of v with respect to the bounds is then ambiguous when $\hat{v}$ is within N_v of one of the bounds (see Fig. 4-6).

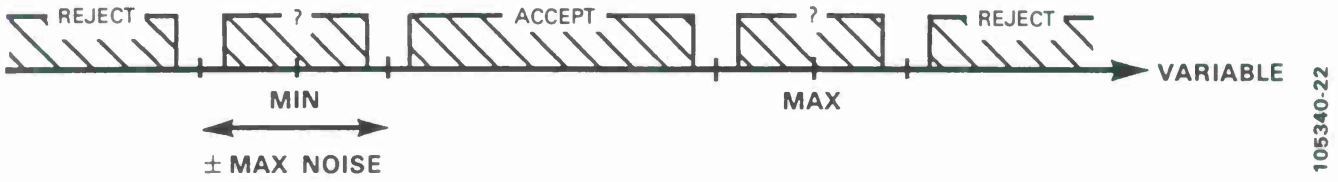


Figure 4-6. Decision regions, estimate with additive noise.

When the estimate $\hat{v}$ is in the "ambiguous regions" in the above figure, the true value of v can be inside or outside the bounds, depending on the particular sample of the additive noise. When testing a sufficient constraint, the ambiguous regions are merged with the reject region because they cannot guarantee a valid value of v. However, when testing a necessary constraint, the ambiguous regions are merged with the accepted region because they could correspond to a valid value of v (see Fig. 4-7).

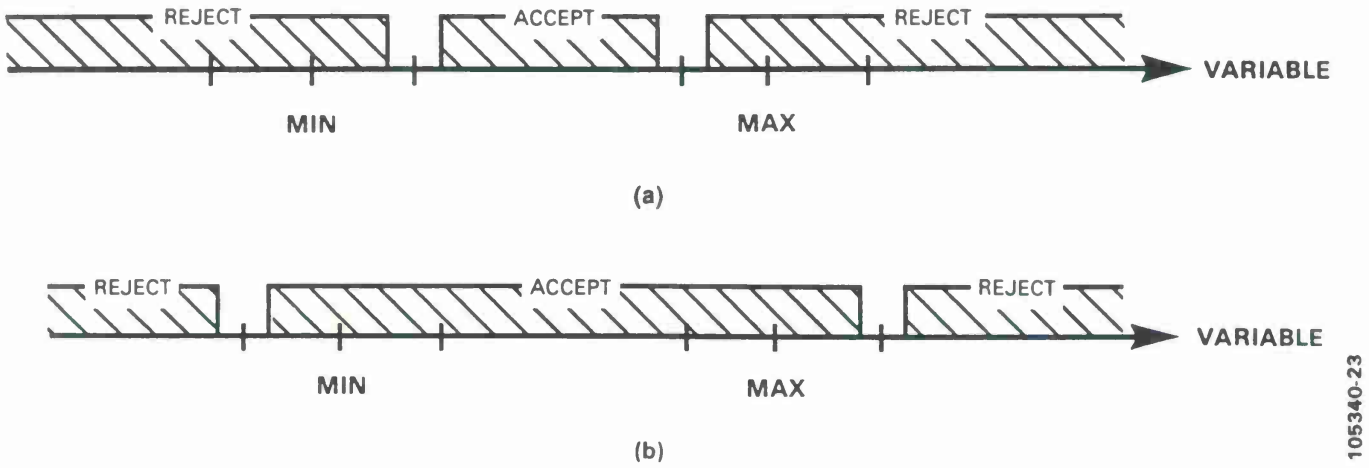


Figure 4-7. Decision regions, noisy estimate, necessary and sufficient tests.

We are interested here only in necessary constraints; the algebraic equations for the test are

$$\begin{cases} \hat{v} + N_v > v_{min} \\ \hat{v} - N_v < v_{max} \end{cases} \tag{4.7}$$

which are equivalent to

$$\begin{cases} Max_{noise}\, v > v_{min} \\ Min_{noise}\, v < v_{max} \end{cases} \tag{4.8}$$

where $Max_{noise}v$ denotes the maximum of the true value of the variable v, given the noisy estimate $\hat{v}$.

The next case to consider is the test of a set of N variables v_i with the bounds v_{min} and v_{max}, first in the absence of noise. The test is the same as Eqn. 4.6 for each of the v_i, and these N tests can be combined into Eqn. 4.9. See Fig. 4-8.

$$\begin{cases} Min_i \hat{v}_i > v_{min} \\ Max_i \hat{v}_i < v_{max} \end{cases} \tag{4.9}$$

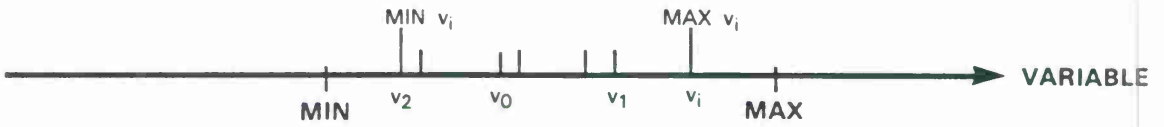


Figure 4-8. Decision test, multiple data, noiseless estimates.

The final case addresses the combination of multiple data and noise. It considers the test of the N variables v_i with the bounds when the estimates $\hat{v}_i$ are corrupted by additive noise of maximum amplitude N_v; the test is a necessary constraint. Given any estimate $\hat{v}_i$, it is only known that the true value of the variable is in the interval $[\hat{v}_i - N_v, \hat{v}_i + N_v]$ (see Fig. 4-9).

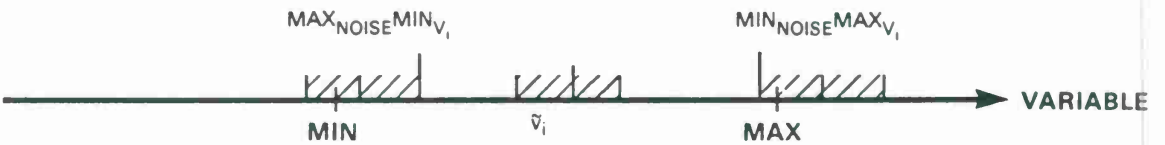


Figure 4-9. Decision test, multiple data, noisy estimates.

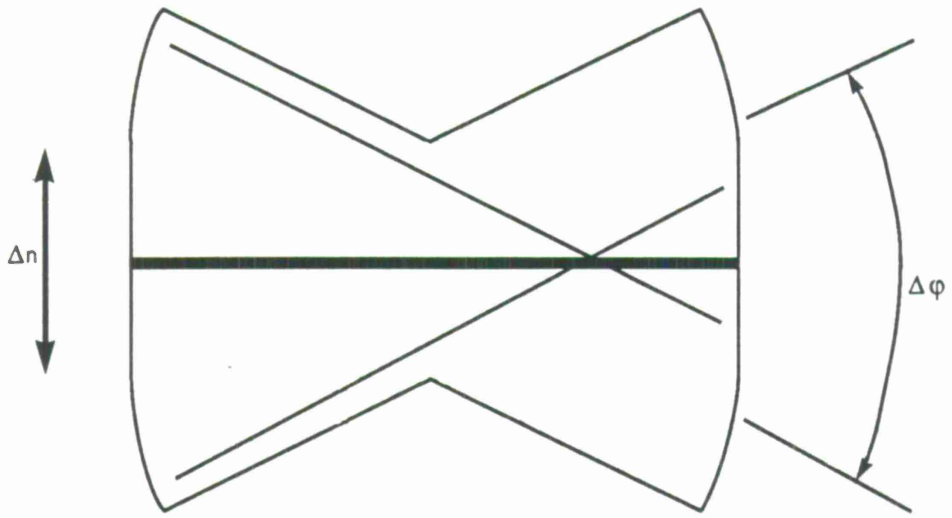
In this case, the necessary constraint tests are given by

$$\begin{cases} Max_{noise}\, Min_i\, \hat{v}_i > v_{min} \\ Min_{noise}\, Max_i\, \hat{v}_i < v_{max} \end{cases} \tag{4.10}$$

Noise Model for Silhouette Edges

The noise model used in our implementation represents both the uncertainty in the orientation of the edges extracted from the image, and the uncertainty in their lateral position. The maximum deviation of the edge from its estimated position is accounted for by the maximum angular deviation $\pm\Delta\phi$ and the maximum lateral deviation $\pm\Delta n$. Errors on the length and longitudinal position of the edge are not considered in our implementation; these are largely offset by the tendency of the silhouette parser to produce silhouette edges shorter than the actual ones.

Figure 4-10 illustrates the range of positions that can be covered by the true edge, given an edge estimate and bounds on errors on its orientation and lateral position.



105340-26

Figure 4-10. Range of positions of the true edge, given a noisy edge estimate.

Configuration Tests for Noisy Edges

The configuration test of a pair of edges, given noisy measurements, consists of testing the ranges of values for the relative orientation ϕ, and for the relative distances t and n. The test combines Eqn. 4.3 to account for potential occlusions, and Eqn. 4.10 to account for possible discrepancies between the true values of ϕ, y, and n, and their estimates from the image data. The tests are given by

$$\left[\begin{array}{l} Min_{noise}\, Max_{sil}\, t_{sil} < Max_{mod}\, t_{mod} \\ Max_{noise}\, Min_{sil}\, t_{sil} > Min_{mod}\, t_{mod} \\ Min_{noise}\, Max_{sil}\, n_{sil} < Max_{mod}\, n_{mod} \\ Max_{noise}\, Min_{sil}\, n_{sil} > Min_{mod}\, n_{mod} \\ Min_{noise}\, \phi_{sil} < \phi_{mod} \\ Max_{noise}\, \phi_{sil} > \phi_{mod} \end{array}\right. \tag{4.11}$$

In the above formulas, $Min_{noise}t_{sil}$ denotes the minimum real value of the variable t at hand, which is compatible with a noisy estimate of the variable extracted from the image. $Max_{sil}t_{sil}$ denotes the largest value of t_{sil} for all the pairs of points on the two estimated silhouette edges. The extrema are attained when each point is at one of the extremities of its edge, and when both the orientation and position noises take their largest values, positive or negative.

4.4 CONSTRAINTS IN THE PRESENCE OF SCALE UNCERTAINTIES

In almost any practical circumstance, the scale of the image is not known exactly. In particular, when the scale of the image is estimated from measurements of the range of the objects in the scene, inaccuracies in the range measurements inevitably translate into inaccuracies in the

estimate of the image scale. Even when the scenes are two-dimensional, there is always some uncertainty in the calibration of the image scale.

Given the uncertainty in the estimate of the scale of the input image, the scale of the configuration of a pair of edges extracted from the image is known only up to the fraction s. Therefore, when the distances between two edges are estimated as $\hat{t}$, $\hat{n}$, the actual distances could have values between $(1-s)\hat{t}$ and $(1+s)\hat{t}$, and $(1-s)\hat{n}$ and $(1+s)\hat{n}$. The effect of scale uncertainties is very closely related to the effect of noise on silhouette measurements; it can be accounted for with the same formalism developed for noise. The constraints including scale variations are

$$\begin{cases} Min_{scale}\, Min_{noise}\, Max_{sil}\, t_{sil} < Max_{mod}\, t_{mod} \\ Max_{scale}\, Max_{noise}\, Min_{sil}\, t_{sil} > Min_{mod}\, t_{mod} \\ Min_{scale}\, Min_{noise}\, Max_{sil}\, n_{sil} < Max_{mod}\, n_{mod} \\ Max_{scale}\, Max_{noise}\, Min_{sil}\, n_{sil} > Min_{mod}\, n_{mod} \\ Min_{noise}\, \phi sil < \phi_{mod} \\ Max_{noise}\, \phi sil > \phi_{mod} \end{cases} \tag{4.12}$$

Although tolerances to scale and to noise errors are implemented with the same formalism, the effects on the constraints are quite different. Noise errors are limited in their absolute magnitude, whereas scale uncertainties produce errors proportional to the distances considered. In the above equations, taking the minimum Min_{scale} is equivalent to multiplying by $(1-s)$; the maximum is equivalent to multiplying by $(1+s)$. Note that the angle constraint is not affected by scale.

4.5 VIEWPOINT-INDEPENDENT CONSTRAINTS

We address now the crucial step of the configuration constraints for pairs of edges, namely the design of constraints that are independent of viewpoint. To make the constraints in Eqn. 4.12 valid over all viewpoints, it is necessary to consider the extrema of the bounds on the right hand sides, for all possible viewpoints. In other words, it is necessary to predict bounds on the configuration of edges in the image, where those bounds are valid over all possible viewpoints.

$$\begin{cases} Min_{scale} Min_{noise} Max_{sil} t_{sil} < Max_{view} Max_{mod} t_{mod} \\ Max_{scale} Max_{noise} Min_{sil} t_{sil} > Min_{view} Min_{mod} t_{mod} \\ Min_{scale} Min_{noise} Max_{sil} n_{sil} < Max_{view} Max_{mod} n_{mod} \\ Max_{scale} Max_{noise} Min_{sil} n_{sil} > Min_{view} Min_{mod} n_{mod} \\ Min_{noise} \phi sil < \phi_{mod} \\ Max_{noise} \phi sil > \phi_{mod} \end{cases} \tag{4.13}$$

As we will show by two simple examples, two edges considered as sticks in three dimensions can appear in very different configurations in the image, depending on the viewpoint; as a consequence, constraints such as Eqn. 4.13 are very weak in that case. However, the recognition system is concerned with solid objects and with observations on the silhouette of the object in the image. This can be exploited to determine the limited set of viewpoints from which a given pair of

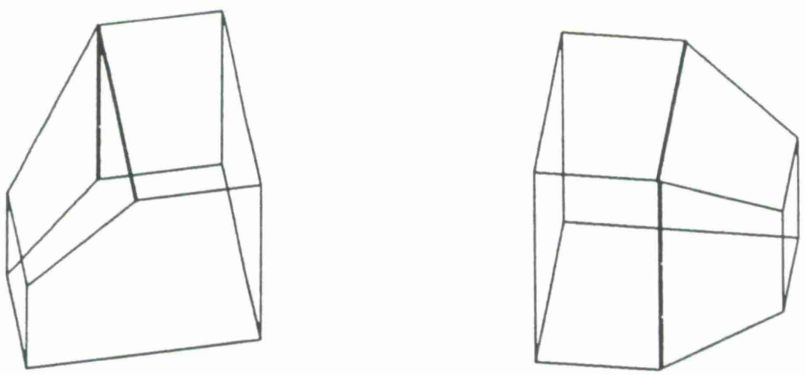
model edges will both appear on the silhouette. This set is often a small fraction of the set of all viewpoints, and the configuration of the pair of edges in the image is much more constrained with this restricted set of viewpoints.

We will first show two simple examples where the natural limitation of the set of viewpoints substantially increases the discriminating power of a configuration constraint. Then, we will analyze the issue of determining which viewpoints map a pair of edges onto the silhouette. Finally, we briefly discuss two methods for evaluating bounds on viewpoint-independent configuration constraints.

4.5.1 Simple Examples of the Use of Visibility

When two edges are considered as sticks in 3-D, the configuration of their projection in the image can vary widely from one viewpoint to the next. For example, two edges with any relative orientation in 3-D can be projected into two edges with any other relative orientation in 2-D, except that parallel 3-D edges remain parallel in the image. In terms of distances, the projections of any two edges will intersect for some viewpoints, and on the other hand, the maximum distance between two points on the edges can be made as large as that distance in 3-D for some viewpoints.

However, when considering 2 edges on a solid object and their projections from the viewpoints for which they appear *on the silhouette*, the configurations of these edges is generally far more restricted. We will illustrate this point with two examples: the first example illustrates this point for angles; the second, for distances.



105340-27

Figure 4-11. Relative angle of two edges of a wireframe object.

Consider the object depicted in Fig. 4-11, and more specifically, the two bold edges drawn on the object. As illustrated in this figure, when the object is considered as a wireframe, the angle between the projections of these edges can range from almost zero to almost 180 degrees. However, when the object is considered as a solid and when the two edges are required to appear on the silhouette of the object, the angle between these edges can only range from 90 degrees to 120 degrees; the images of the object for viewpoints close to these extrema are shown in Fig. 4-12.

105340-28

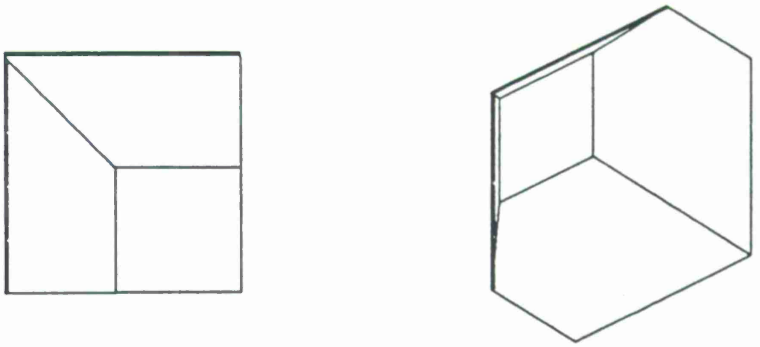


Figure 4-12. Relative angle of two edges of an opaque object.

105340-29

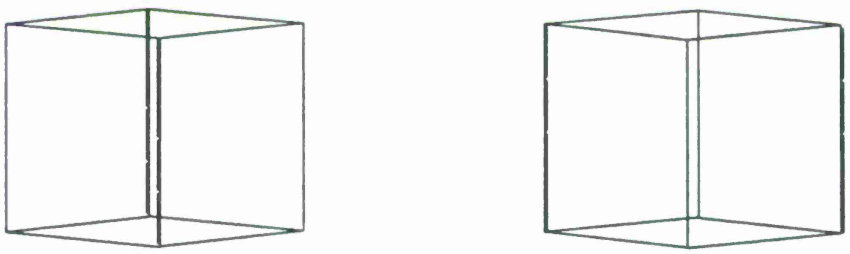


Figure 4-13. Distance between two opposite edges of a wireframe cube.

As another example, consider the unit-side cube illustrated in Fig. 4-13 and more specifically, the distance between two opposite edges of the cube, such as those emphasized in bold on the figure. With the cube considered as a wireframe, the distance between the projections of the two edges can have any value between 0 and $\sqrt{2}$. However, when the cube is considered as opaque, the distance between two edges can only vary between 1 and $\sqrt{2}$, as illustrated in Fig. 4-14.

105340-30

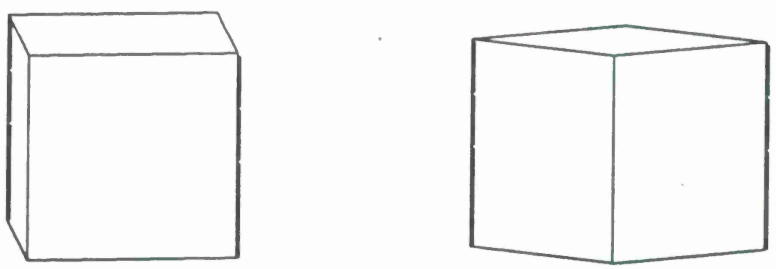
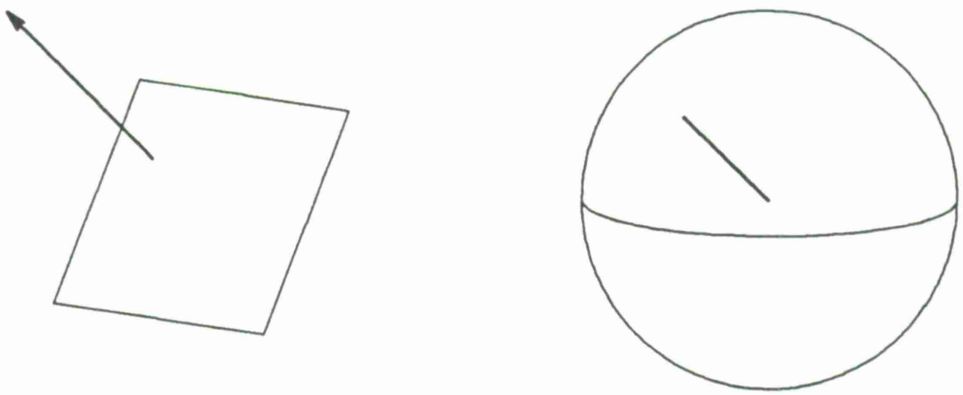


Figure 4-14. Distance between two opposite edges of a solid cube.

In the above examples, it is clear that much tighter bounds can be set on the range of configurations of a pair of edges when the visibility criterion is used. In the following subsection, a theoretical basis for the visibility criterion is presented.

4.5.2 Visibility of a Pair of Edges on the Silhouette

In this section, we will analyze the set of viewpoints for which a pair of edges appears on the silhouette, and more specifically, we will address this question for the pair of edges considered previously on the solid of Fig. 4-12. We will first determine the set of viewpoints for which one edge appears on the silhouette, then the viewpoints for which both edges appear simultaneously on the silhouette. In the discussion of viewpoints, we will represent viewpoints by points on a unit sphere, where a point on the sphere corresponds to the viewing direction parallel to the vector from the origin to the point, as in Fig. 4-15.

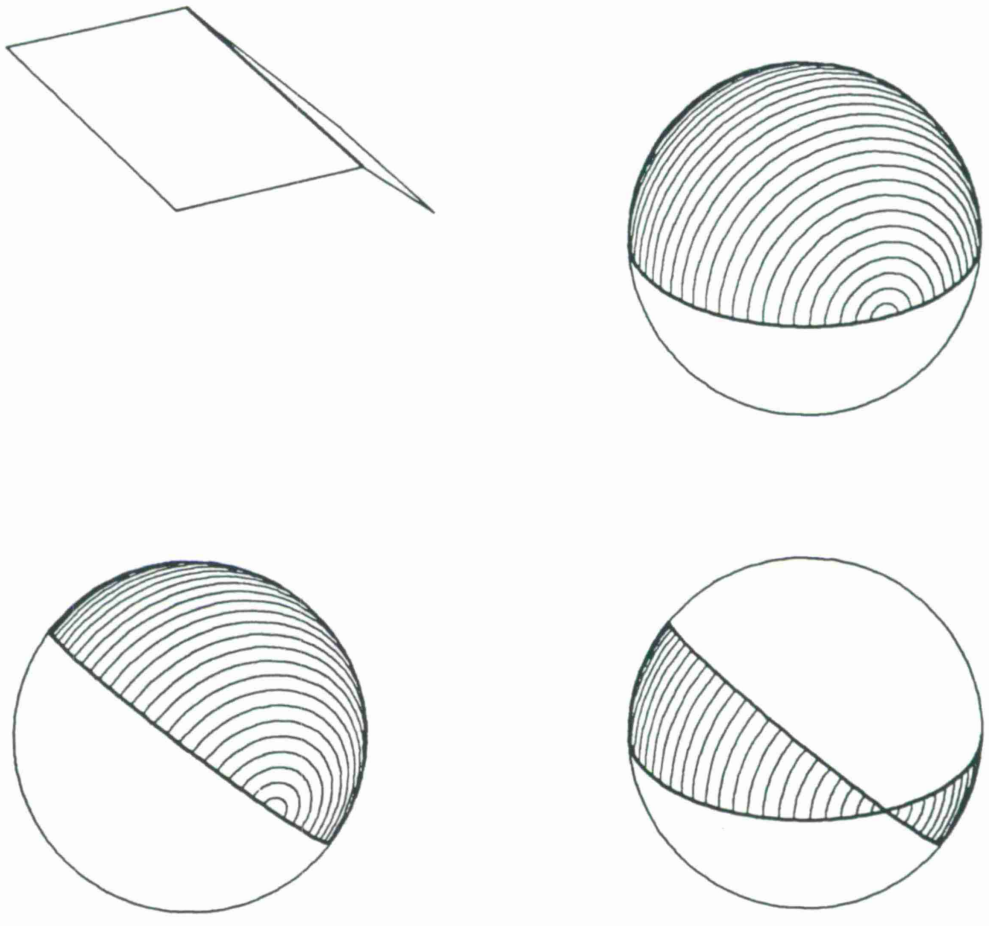


105340-31

Figure 4-15. Representation of viewpoints as points on a unit sphere.

Consider first the edge E_1 in Fig. 4-16(a). This edge may appear on the silhouette only if the adjacent face F_1 is visible and the other adjacent face F_2 is hidden, or vice-versa. The set of viewpoints for which a face, say F_1 is visible, is the hemisphere H_1 whose pole corresponds to the normal to the face, see Fig. 4-16(b). Similarly, the face F_2 is visible for viewpoints on the hemisphere H_2 corresponding to its own normal orientation, as shown in Fig. 4-16(c). The viewpoints for which one and only one of these faces is visible is the symmetric difference (X-OR) between H_1 and H_2, which is an area in the shape of two crescents on the unit sphere, see Fig. 4-16(d). The same analysis can be applied to show that the edge E_2 appears on the silhouette only for the viewpoints illustrated in Fig. 4-17. Both edges E_1 and E_2 will appear simultaneously on the silhouette for the viewpoints at the intersection of the regions defined for E_1 and E_2 separately; as illustrated in Fig. 4-18.

Note that visibility regions are always symmetrical with respect to the center of the sphere; the visibility region for E_1 and E_2 is composed of the quadrangle visible in Fig. 4-18 and of a similar quadrangle in the back of the sphere. This symmetry is consistent with the observation that silhouettes obtained for opposite viewing directions are identical except for a mirror symmetry. In the sequel, we will often consider only one of those symmetric sets of viewpoints. The set of viewpoints for which both edges appear simultaneously on the silhouette is a spherical polygon in general. In the absence of particular alignments of the faces adjacent to the two edges, the region is a spherical quadrangle, which may be convex as in Fig. 4-18 or self-intersecting as in Fig. 4-19. When two of the four faces adjacent to the two edges are parallel, the spherical polygon is a

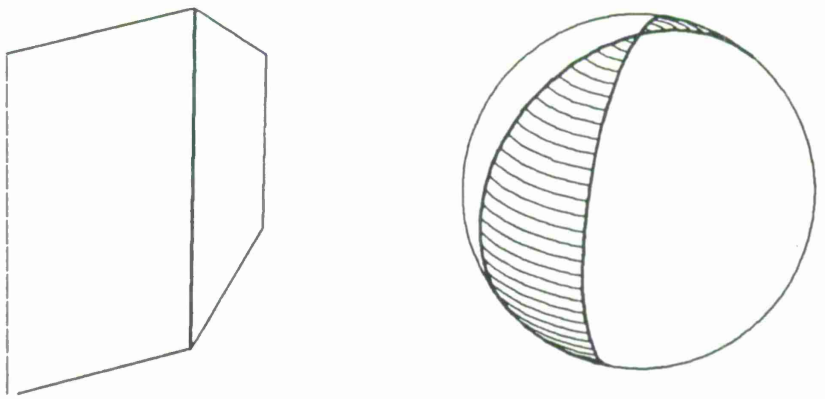


105340-32

Figure 4-16. Visibility of an edge on the silhouette: (a)edge, (b)visibility of face 1, (c)visibility of face 2, (d)edge visibility.

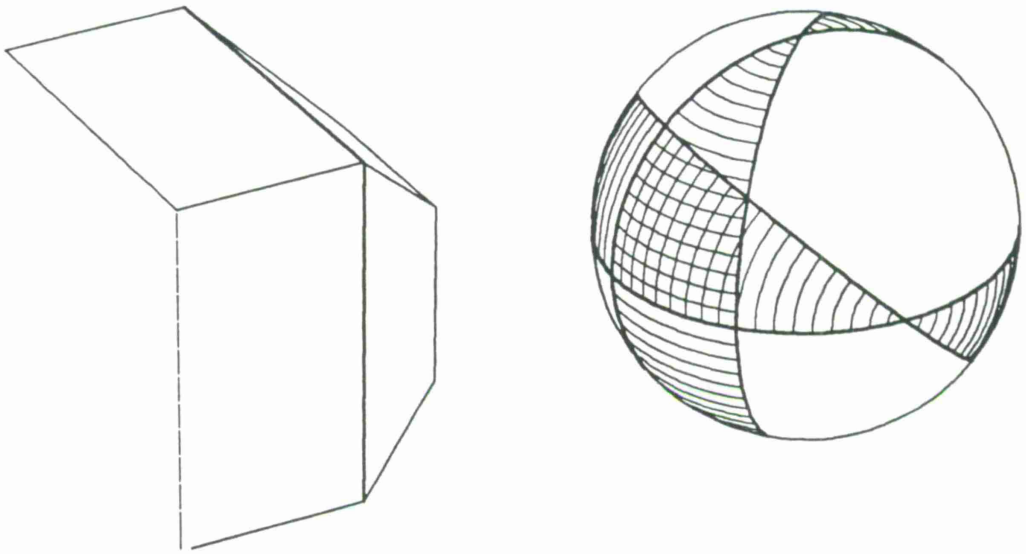
triangle, as in the example of Fig. 4-20. Finally, when E_1 and E_2 are parallel, the visibility region reduces to two crescents such as in Fig. 4-16(b); this is the case for the pair of edges considered on the cube of Fig. 4-13.

In the above discussion, the viewpoints for which a model edge appears on the silhouette were determined solely on the basis of occlusion by the faces adjacent to the edge. For some non-convex objects, the above method will determine a visibility region for a given convex edge, but the edge itself will never appear on the silhouette, due to occlusions by remote parts of the object. This would clearly be the case for internal details in an object in the shape of a box such as the one illustrated in Fig. 4-21. It is extremely difficult to address the occlusion by non-adjacent faces analytically; however, visibility regions determined by the above method can only be overestimated so that the thresholds evaluated for the models with these regions may only err in being too weak.



105340-33

Figure 4-17. Visibility of the second model edge on the silhouette.

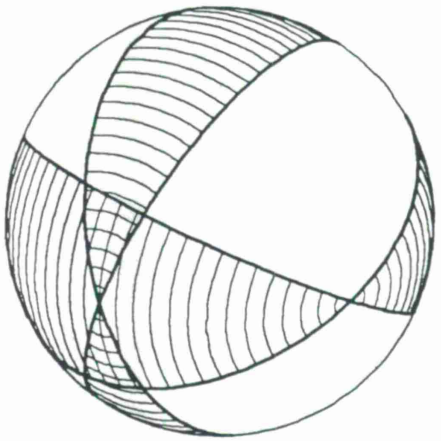


105340-34

Figure 4-18. Visibility of the pair of model edges on the silhouette.

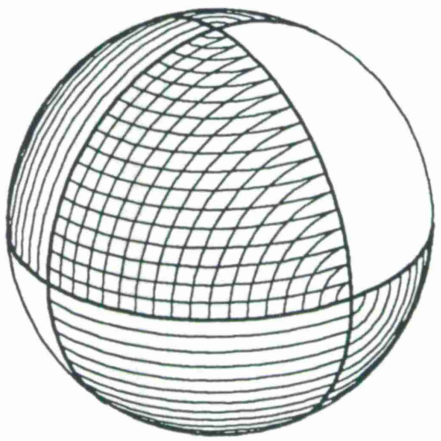
4.5.3 Viewpoint-Independent Constraints

We address now the evaluation of the bounds on the right side of Eqn. 4.13. We have considered two different techniques for evaluating the extrema over the set of viewpoints for which a pair of model edges appears on the silhouette. The first technique is based on the visibility analysis described in the previous section; the problem is one of finding extrema of functions specifying the configurations of model edges projected in the image, given inequality constraints restricting the viewpoint to a spherical polygon on the sphere. This technique will be referred to as the analytical technique. The second technique, which will be referred to as the "brute-force" technique, consists of evaluating a large number of silhouettes of the object and of computing the extrema of the configuration over all these silhouettes.



105340-35

Figure 4-19. Visibility of a pair of edges: concave spherical quadrangle.



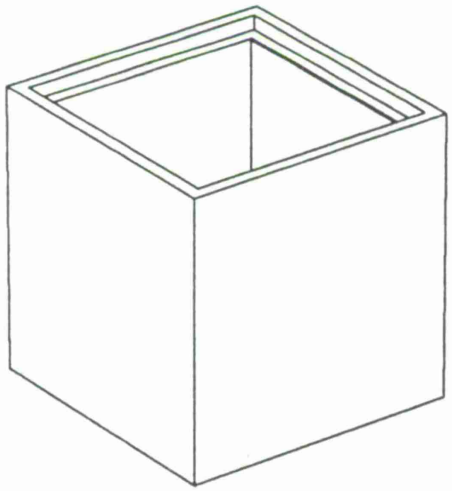
105340-36

Figure 4-20. Visibility of a pair of edges on the silhouette: special alignment.

Analytical Method

The first issue in the analytical method is to determine and represent the set of viewpoints for which the pair of edges appears on the silhouette. The crescents corresponding to each edge can be represented by spherical polygons with two edges; their intersection can be determined as a general intersection of two spherical polygons. It is advisable to devise the framework in terms of general spherical polygons so that the system can also compute intersections with other sets of viewpoints, for example with regions describing a-priori information on viewpoint.

Once the visibility region has been determined as a spherical polygon, the computation of the extrema on the right side of Eqn. 4.13 corresponds to the estimation of an extremum with inequality constraints. The values of t_{mod} and n_{mod} specifying the configuration of edges in the image can be determined by vector algebra from the 3-D configuration of the model edges and in terms of the viewpoint. For a convex visibility region, the inequality constraints correspond



105340-37

Figure 4-21. Invisible edge declared visible by the local analysis.

to setting the viewpoint towards the inside of each edge of the spherical polygon. The extremum versus viewpoint can be determined by the Kuhn-Tucker formalism.

Brute-Force Method

In the brute-force method, silhouettes of the model are determined for a large set of viewpoints, corresponding to points evenly spread across the sphere. The edges of each silhouette are marked to retain the identity of the corresponding model edge. The values of $Min_{mod}t_{mod}$ and of other similar distances are evaluated for each pair of edges on each silhouette. Extrema of these distances are then evaluated over all the silhouettes, for each pair of model edges.

With this method, the viewpoint issue is addressed implicitly when computing each silhouette. Indeed, a given pair of edges will appear only on the silhouettes with viewpoints in the appropriate visibility region. For other viewpoints, the distances will be noted as nonexistent. It is interesting to note that with this method, visibility is addressed not only with respect to the faces adjacent to the edge, but with respect to the entire object. Since the set of viewing directions is only sampled, the extrema obtained with this method are underestimated in general.

Comparison of the Two Methods

The two methods presented above both have advantages and disadvantages. We have opted for the second method because of a number of advantages including the ease and extensibility of its implementation, and the stronger constraints that it provides.

The implementation of the analytical method requires the determination of the configuration of a pair of image features as a function of the configuration of their counterparts in the 3-D model and as a function of the viewpoint. In addition, extrema of these configurations, both free and

with constraints, must be analyzed. The analytical work is quite tedious; the implementation is also quite involved and is different for each characterization of the configuration. On the other hand, the brute-force method requires methods for estimating configurations from a silhouette only; these algorithms must be implemented for the run-time system anyway. The computation of all the silhouettes is done once, and the evaluation of the extrema is independent of the variable being considered. Note that when evaluating the extrema, only one silhouette and its configuration must be stored at any given time; these can be discarded after the current estimates of the extrema are updated. For objects of interest having tens of edges, we have observed that the execution times for the two methods are on the same order of magnitude, although these times increase as the square of the number of edges for the analytic method, whereas the increase for the the brute-force method is dominated by the time to evaluate silhouettes, which has a slower increase as a function of the number of edges. To summarize the above discussion, the implementation of the constraint threshold prediction is favorable to the brute-force method.

If different image features are added to the recognition system, or if other measurements of feature configurations are chosen, in both cases, it is necessary to implement the estimate of new constraints for 2-D input data. However, with the analytic method, it is also necessary to implement the estimate of the new constraints for 3-D model data; this is in general much more demanding than the 2-D case. As a consequence, extensions of the strategy are easier to implement with the brute-force method.

The final category in which we compare the two methods is that of accuracy. The analytic method does compute exact maxima of the thresholds, for the problem that it solves; however, as we saw earlier, the visibility region determined by the analytic method can be too large. As a result, the constraints will be exact in some cases, but may be looser than necessary in other cases. The brute-force method samples the set of viewing direction so that it is unlikely to produce the exact thresholds for the constraints. The errors due to the sampling are difficult to evaluate, and since they underestimate the threshold bounds, they could result in correct matches being rejected. Although this is a serious problem, the errors can be reduced by performing a fine sampling of the set of viewing directions (2400 points in our implementation); the effects of these errors are also reduced by the noise margins tolerated for all edge measurements.

To summarize the comparison of the two methods for compiling the object, the analytic method has the advantage that it adheres perfectly to the strategy of necessary constraints. Although the brute-force method can be failed on that criterion, the consequences of this failure can be minimized, and are vastly offset by advantages on all other points. The system presently compiles the constraint thresholds for the models by the brute-force method.

5. 3-D OBJECT MODELS, 2-D SILHOUETTES

In this section, the internal representation of 3-D objects and 2-D silhouettes is discussed. The material covered in this section is largely related to implementation issues, but it is important to study this component to develop a sound understanding of the recognition system.

5.1 3-D OBJECT MODELS

As we mentioned earlier in this report, all objects are described by rigid 3-D polyhedra in our system; these can either reflect the exact shape of the 3-D objects or an approximation of their shape in the case of curved objects. A complete description of a polyhedron consists of sets of all its components, i.e., vertices, faces and edges, together with all the connectivity relations among the components. However, as we will see in this section, the only requirement set on the polyhedral model is the ability to generate synthetic silhouettes. This requirement can be satisfied by a simplified description of the polyhedron geometry. Furthermore, the user interface can be simplified by providing simpler description languages and compiling these descriptions into the geometric representation in terms of vertex, edge and face components.

In this section, we will first discuss the representation of polyhedron geometries, then the synthesis of silhouettes given the geometric representation, and finally the compilation of edge constraints.

5.2 DESCRIPTIONS OF POLYHEDRA

Polyhedra are solids bound by planar faces; the system considers convex polyhedra, concave polyhedra, polyhedra with handles and non-connected polyhedra. However, we exclude self-intersecting polyhedra since they do not model solid objects, and "hollow" polyhedra since those cavities have no impact on silhouettes. Polyhedra are usually defined in terms of their vertices, edges and faces. The numbers of these components, noted V, E, F, are constrained by a relation due to Euler,

$$V + F - E = 2(S - H) \tag{5.1}$$

where S represents the number of disconnected pieces of the solid and H the number of handles. In the above formula, it is assumed that faces have no holes. Otherwise, the total number of holes in faces must be subtracted from the left side. Many polyhedron description systems perform tests on the input data to verify the connectivity of components, which in the end ensure the validity of Eqn. 5.1. In our system, a mixed approach is taken, in that the validity of the input data is verified for connectivity and hence implicitly for the validity of Eqn. 5.1, but the stored representation is simplified by removing some parts such as "flat" edges. These flat edges have the same face on both sides, and are often inserted by solid modelers to connect the boundary of

a hole in a face with the outer boundary of the face. In brief, the system verifies the input data but builds a stored representation which may be inconsistent.

The basic input description of a polyhedron in our system is a list of vertex geometries and a list of the vertices on each face; with this information, it is possible to determine the edges of the solid and all the connectivities. More specifically, each vertex is represented in the input as a list of its three coordinates (xyz); the set of vertices is input as a list of those coordinate lists, where the index of each vertex in the list uniquely specifies each vertex. Each face is represented by a list of the indices of the vertices around the contour of the face. The contour of the face is listed counterclockwise when looking at the face from the exterior of the object.

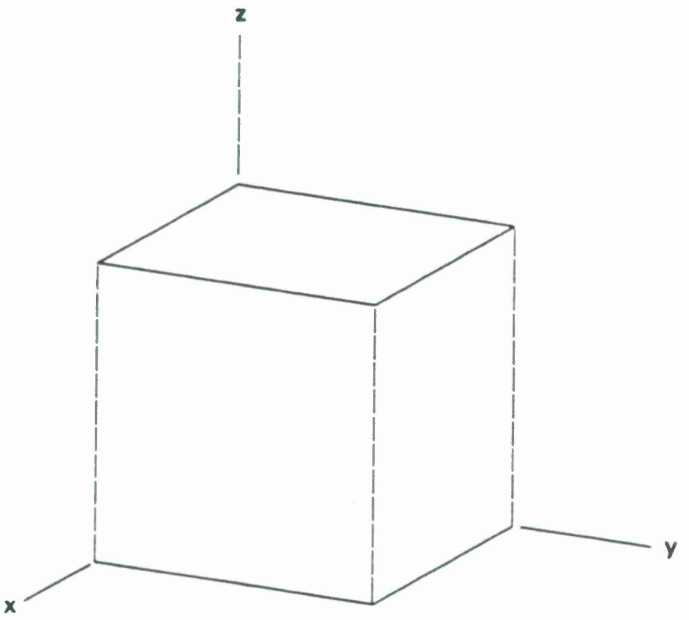
```
(define-object
  'cube
  :vertices '((0 0 0) (0 0 1) (0 1 0) (0 1 1)
              (1 0 0) (1 0 1) (1 1 0) (1 1 1))
  :faces '((0 1 3 2) (0 4 5 1) (4 6 7 5)
        (2 3 7 6) (1 5 7 3) (0 2 6 4))
  )
```

Figure 5-1. Definition of a cube.

An example definition of a cube is shown in Fig. 5-1 where the coordinates indicate that the cube is in the first octant and has unit side. The first face in Fig. 5-1 has vertices 0, 1, 3, 2 along its perimeter, which is the side facing the negative x-axis (see Fig. 5-2). Clearly, it is not possible to explicitly define a face with holes with the above representation; however, holes in a face can be eliminated by connecting them with the outside contour of the face, see the example in Fig. 5-3. Given an object definition such as Fig. 5-1, the system builds a LISP object containing three arrays: one each for the vertices, the faces and the edges of the system. Each vertex is a 3-D point structure recording its 3-D coordinates; each face is a structure recording the equation of the face plane and the list of points around the face; each edge is a structure recording both endpoints and both adjacent faces. The structures have slots for recording extra attributes such as bounding boxes for faces and edges, and the visibility of a face.

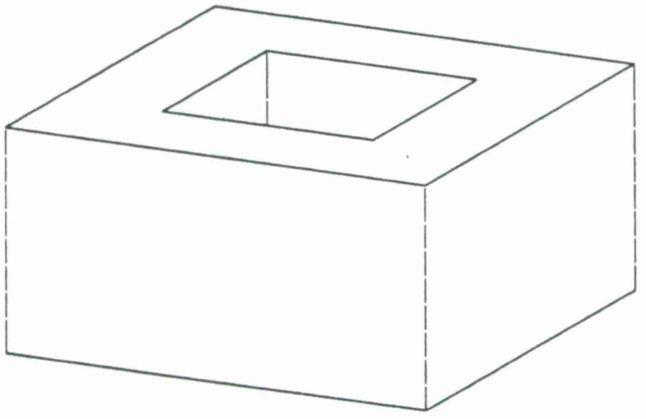
5.2.1 Constructive Geometry

Although it is possible to enter any polyhedral shape by a definition such as the one shown in Fig. 5-1, it is clear that this method is quite tedious and error-prone for the definition of complex polyhedra. To generate most of the models in this report, we have used YASM, a constructive solid geometry (CSG) package developed at MIT by Alain Lanusse. In YASM, solids are defined in terms of simple primitives and their relations. As an example, the table model shown in Fig. 5-5 was defined by the expression in Fig. 5-4.



105340-39

Figure 5-2. Cube as defined above.



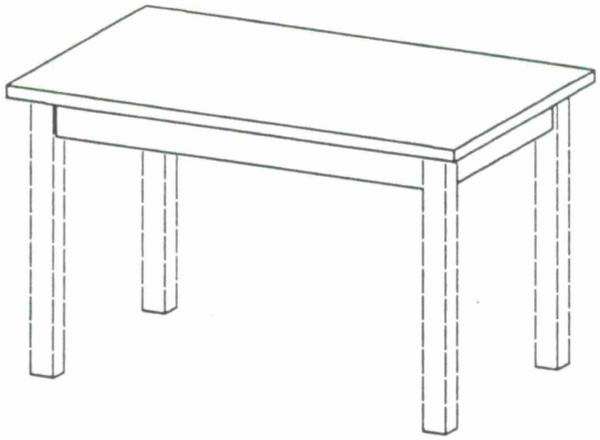
105340-40

Figure 5-3. Example of a solid with a hole in a face.

The primitives provided by YASM include boxes and approximations of curved surfaces such as cylinders, cones, spheres and tori. These primitives can be moved by arbitrary translations and rotations, and they can be combined by intersection, union or difference. The internal representation of objects in YASM describes their vertices, faces and edges. The YASM system was very useful in designing object models; however, it has its disadvantages. For example, the system cannot handle combinations of objects if there are any particular alignments.

```
(define-yasm-solid
  ;; dimensions are in feet
  'table
  (model-union
    ;; table top
    (model-translate (make-box 2.5 4.0 0.10) 0.0 0.0 -0.05)
    ;; table legs
    (model-translate (make-box 0.2 0.2 2.35)  1.0  1.75 -1.25)
    (model-translate (make-box 0.2 0.2 2.35) -1.0  1.75 -1.25)
    (model-translate (make-box 0.2 0.2 2.35)  1.0 -1.75 -1.25)
    (model-translate (make-box 0.2 0.2 2.35) -1.0 -1.75 -1.25)
    ;; table beams, long side
    (model-translate (make-box 0.1 3.5 0.35)  1.0 0.0 -0.25)
    (model-translate (make-box 0.1 3.5 0.35) -1.0 0.0 -0.25)
    ;; table beams, short side
    (model-translate (make-box 2.0 0.1 0.35) 0.0  1.75 -0.25)
    (model-translate (make-box 2.0 0.1 0.35) 0.0 -1.75 -0.25)
    )
  )
```

Figure 5-4. Table definition in yasm.



105340-42

Figure 5-5. Table model generated by yasm with the above definition.

5.2.2 Generating Silhouettes of Objects

A major requirement of the solid models is that they must be able to provide silhouettes of the object for any given viewpoint. It is well known [25] that the silhouette of a convex polyhedral

object is the projection of the silhouette generator of the polyhedron for the given viewpoint. The silhouette generator itself is the set of edges adjacent to both a face visible from the viewpoint and a face hidden from the viewpoint. In the case of a non-convex polyhedron, the silhouette is obtained in the same way, except that parts of the projection of the silhouette generator may be occluded. The algorithm used to generate the silhouette first generates candidate silhouette edges, namely the projection of the silhouette generator edges, and then clips these to take occlusions into account.

To generate the candidate silhouette edges, the faces of the polyhedron are first investigated, and their planes are compared with the viewpoint. The faces are marked as visible or hidden. In the next step, each edge of the polyhedron is considered; the silhouette generator edges are those which are adjacent to one visible face and one hidden face. Finally, the candidate silhouette edges are obtained by projecting the silhouette generator edges onto the image plane. The silhouette itself is obtained by clipping the candidates to account for occlusions; the clipping is done by the projection of all the visible faces of the polyhedron. In the projected silhouette, each edge retains the index of the corresponding model edge; therefore, it is possible to relate each edge of the synthetic silhouette to the corresponding model edge.

We will see in the next subsection that it is crucial to optimize the speed of the silhouette generation; we have implemented a number of strategies to increase the speed of both parts of the silhouette synthesis process, namely the projection of candidate edges and the clipping.

One strategy we have adopted improves the efficiency of silhouette generator projection by keeping two copies of each model, a master copy and a work copy. Both instances have exactly the same structures and links; the projection is operated by simply altering the coordinates of the vertices and planes in the structures of the work copy. This procedure has the advantage that no new structures need be instantiated and that the links between edges, faces and vertices need not be copied. The only operation required before the projection of the point and plane coordinates is to refresh the state of the work copy by copying the coordinates from the master copy.

The clipping operation consists of retaining all the parts of the candidate silhouette edges which are outside the projected faces of the solid. To improve the efficiency of the clipping operation, we have devised a number of strategies. Very substantial gains are obtained by comparing the bounding boxes of edges and faces before starting the general clipping operation; if the boxes are distinct, no clipping takes place. Another strategy that we have adopted is to subdivide the projection plane into a grid of regions, for example a grid of 10 x 10 regions. For each region, we build a list of the faces whose bounding boxes intersect the region. The clipping operation consists of three phases for each edge. In the first phase, the edge is compared with the grid to determine which grid boxes overlap with the edge bounding box. The faces marked in these boxes are the only faces that can potentially clip the edge. The second phase rejects some of the potential clipping faces by comparing the bounding boxes. Finally, the last phase consists of the exact clipping of the edge by the remaining faces. We have observed substantial gains with the bounding box test in all cases. However, the space subdivision technique becomes advantageous only for relatively large object sizes; for most of our examples, those gains were inconclusive.

5.2.3 Compiling Object Models

A major requirement of the object model in our system is to provide constraints on the configurations of pairs of edges as they may appear in silhouettes of the object. As indicated in the previous section, we have adopted the brute-force method to generate these constraints. Large numbers of silhouettes are generated for evenly spread viewpoints, constraints are evaluated for each silhouette individually, then combined into tables for viewpoint-independent constraints. Note that it is never necessary to store the silhouettes or the corresponding constraint tables. For each viewpoint, the silhouette constraints are computed on the fly, and they are used immediately to update the tables of model constraints. A few additional details regarding the constraint tables are now discussed.

For a given model, the constraint tables must store the values of t^{ij}_{min}, t^{ij}_{max}, n^{ij}_{min}, n^{ij}_{max}, ϕ^{ij}_{min}, ϕ^{ij}_{max} for each pair (ij) of model edges. The values of these variables correspond either to the extrema of the distances or angles of the model edges as they may appear on the silhouette, or to `nil` to indicate that the pair never appears simultaneously on the silhouette. These values are stored in $M \times M$ arrays, where M is the number of model edges. Initially, the model arrays are filled with `nil`'s. For each viewpoint, the synthetic silhouette of the model has S edges, and 6 $S \times S$ tables (a *min* and *max* table for each of t, n, and ϕ) are generated to store the silhouette constraints for that viewpoint. These silhouette tables are used to update the model tables by comparing each element of the silhouette table with the corresponding element in the model table. In order to determine the correspondences between silhouette table elements and model table elements, we use a list of indices characterizing which model edge corresponds to each edge of the silhouette. In the actual implementation of the system, a further refinement has been developed for the tables of tangent distance and angle constraints. In addition to a minimum and maximum value, those tables also store a minimum positive value and a maximum negative value. These additional values provide for increased constraint power when the set of viewpoints is restricted.

A number of advantages of the brute-force compilation of edge-pair constraints were discussed in Section 4. An additional advantage is that a-priori limitations on the viewpoint are very easy to implement. Indeed, to determine the constraints for a limited set of viewpoints, the brute force method is used with viewpoints sampling the restricted set of viewpoints considered. When restricted sets of viewpoints are considered, it is possible for the range of values of angle and tangent distances to be disconnected sets. In particular, when a set of viewpoints includes both "front" views and "back" views, the ranges of t and ϕ are symmetric with respect to 0. We have introduced constraints with 4 tests, namely for both negative and positive minimum and maximum, to exploit the disconnected sets of legitimate values.

5.3 SILHOUETTE REPRESENTATION

The silhouettes of objects in images represent the outline of the corresponding objects in the projection plane. The "raw"silhouette is a chain of points along the contour, or a set of chains for a silhouette with multiple parts. The chain is parsed to produce a set of straight edges representing

prominent parts of the silhouette. Finally, the configuration of these straight edges is abstracted in six constraint tables representing the ranges of distances and angles between points on each pair of edges. Among the three representations of the silhouette, namely the chain, the edges, and the constraint tables, the last two are actively exploited during the recognition algorithm. Therefore, the silhouette is represented at run-time by a structure combining these two descriptions. The chain is used only after the search of interpretations, to provide a measure of confidence in each interpretation. Among the silhouette processing tasks, the extraction of the silhouette chain from the image is performed with well-known techniques which will be reviewed only very quickly. The parsing of the silhouette chain into a set of straight edges is important, and must be performed carefully to provide good edge features for the recognition algorithm. Silhouette parsing methods are discussed in detail in Appendix A. Finally, the setup of constraint tables describing the configuration of pairs of edges closely follows the strategy developed in Section 4.

5.3.1 Silhouette Chain Extraction

Edge chains are extracted from the silhouette image by first convolving the image with a discrete version of the Laplacian of a Gaussian; the edges are the zero-crossings of the resulting image. These zero-crossings are detected and linked by standard image processing algorithms [27]. The result of this process is a list of chains representing the zero-crossing contours of the convolved image. Simple heuristics can be used to discard spurious contours. Edge detection by Laplacian of Gaussian has the advantage that the contours are linked by default, whereas other edge detection schemes, such as the Canny edge operator, produce isolated edge points. However, the Laplacian of Gaussian edge detection produces biases which can be significant at corners. The system correctly handles these errors, but system performance could only be improved by providing better primitives.

5.3.2 Silhouette Parsing

Each silhouette chain is parsed into a description in terms of straight edges. Note that the description need not be complete, but must describe the salient features of the silhouette. An important characteristic of the silhouette parser is that it must carefully distinguish the straight edges from the curved silhouette sections to correctly represent each during the matching procedure. Specifically, the curves are modeled by a set of isolated points and the normal orientation of the silhouette at these points. We have chosen to describe edges by their center point, normal orientation and length; we give the name of "edge element" or edgel, to the edge characterized by this representation. Both finite edges and isolated points with a normal orientation can be represented by edge elements.

The silhouette parser first detects candidate points for straight edges by analyzing estimates of the smoothness and curvature of the silhouette. The candidates are grouped and grown into straight edges. Among the remaining points, candidate points on smooth curves are detected by analyzing the smoothness and curvature, although with different thresholds this time. These points are grouped into smooth curve strings which are modeled by carefully chosen sets of isolated

points. The silhouette parser records with each edgel, an estimate of the errors on the normal orientation and lateral position of the edge. The system uses a graphical representation consisting of the edge itself and an outgoing normal. The length of the normal indicates the confidence in the normal orientation of the edge (see Fig. 5-6).

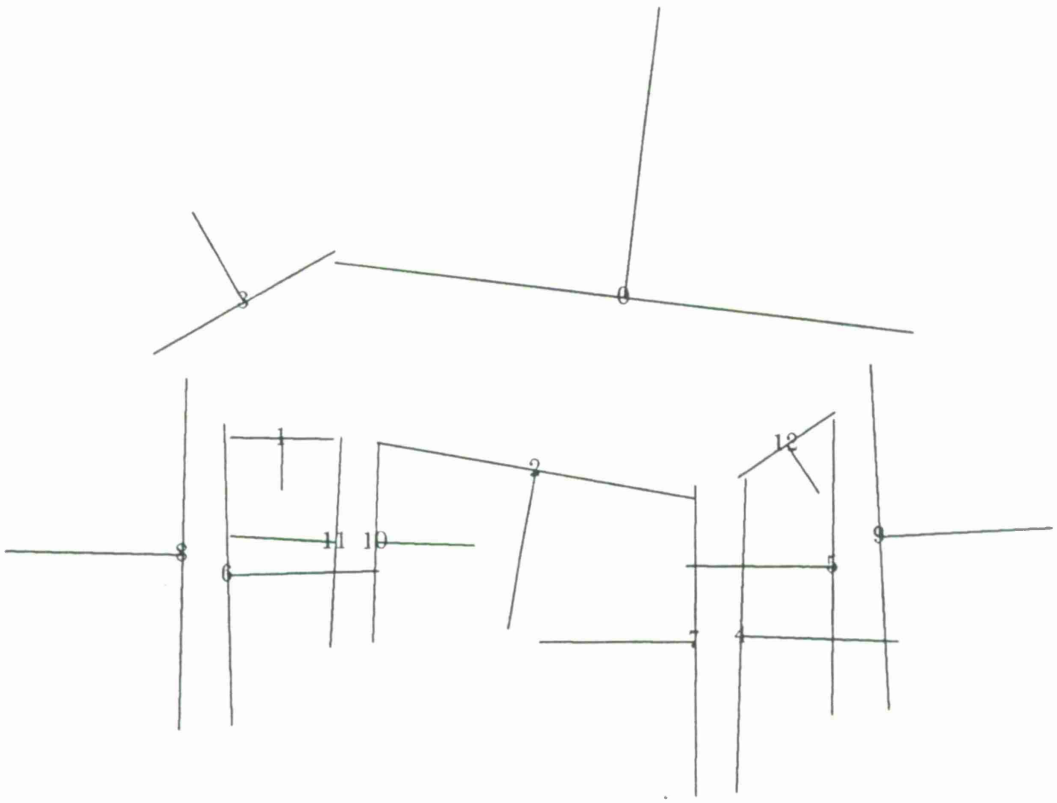


Figure 5-6. Example of the graphical representation of silhouette edgels.

In the current implementation, the estimated error on the normal orientation of edges is smaller for longer edges, as can be seen in Fig. 5-6. Further details on silhouette parsing are provided in Appendix A.

5.3.3 Silhouette Constraints

During the tree-pruning search of interpretations of silhouette edges in terms of model edges, the range of distances and angles of the observed edges are compared to the allowable ranges predicted from the model. In the current implementation of the system, all the constraint ranges are computed for all pairs of silhouette edges before starting the tree search. The corresponding values are stored in six arrays retaining, respectively, the minima and maxima of the normal distance, tangent distance and relative orientation of points on the two edges. The silhouette structure stores both the list of edge elements and these six arrays describing the edge configuration. During

the tree search, the silhouette is accessed only through the six arrays, whereas the verification phase of the recognition algorithm uses the list of edgels to compare the observed silhouette and a silhouette synthesized from the model.

6. TREE SEARCH OF EDGE INTERPRETATIONS

In this section, the organization and pruning of a hypothesis space of edge interpretations is discussed. The hypothesis space is conceptually organized into a tree of all potential interpretations of silhouette edges in terms of model edges. This tree is efficiently pruned by applying simple constraints on the configurations of pairs of interpretations. We will first describe the structure of the interpretation tree, then discuss traversing the tree by a backtracking strategy, and finally expose a number of heuristic methods that can be used to improve the efficiency of the search.

6.1 ORGANIZATION OF THE HYPOTHESIS SPACE

The silhouette edge interpretation problem consists of labeling each edge in a set of S silhouette edges, with a label corresponding to one of M model edges. Without a-priori constraints on the interpretation, each silhouette edge has M possible interpretations, corresponding to the M possible model edges. Since an edge of the 3-D object can appear as split in the silhouette, due to either occlusions or early vision processing artifacts, two or more silhouette edges can be matched to the same model edge (see Fig. 6-1).

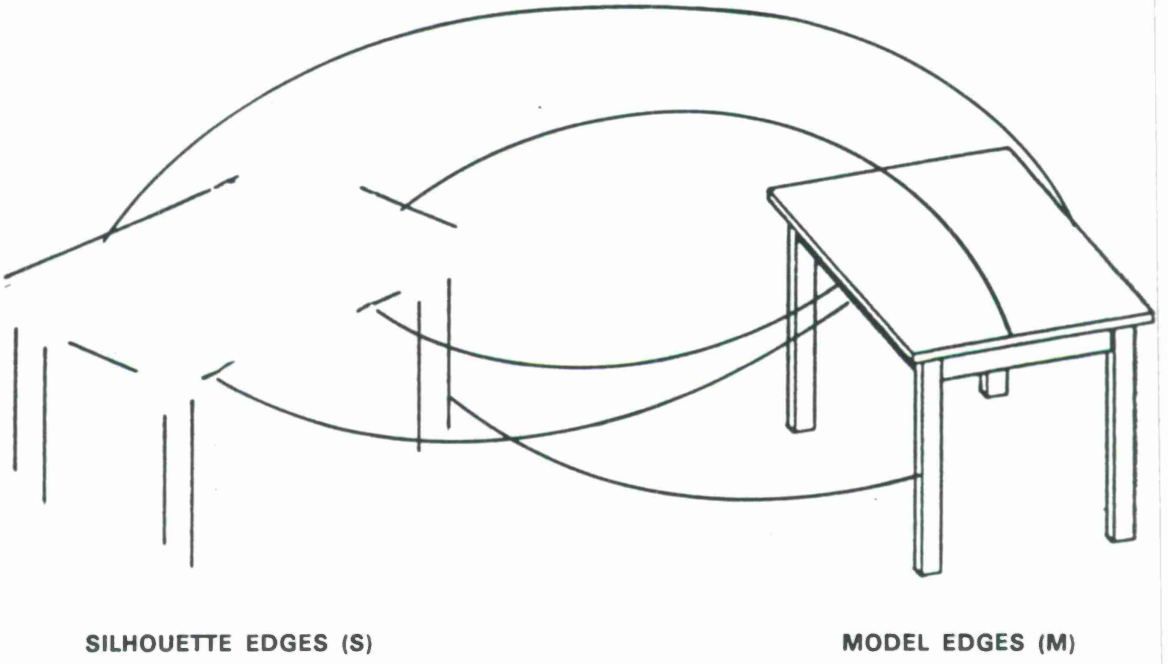


Figure 6-1. Interpretations of silhouette edges in terms of model edges.

Since each of S silhouette edges has M independent interpretations, the total number of silhouette edges interpretations is M^S, which is a very large number in any interesting case. Typical

figures range between 10^{10} and 10^{100}. It is unrealistic to test all these hypotheses by brute force. However, as we will see, this huge set of interpretations can be organized into a tree structure and searched efficiently for legitimate interpretations. The tree search effectively considers the whole set of M^S hypotheses, but with a vastly reduced effort.

In our system, the interpretations of silhouette edges in terms of model edges are conceptually organized into a balanced tree of depth S and branching factor M. A node at level K of the tree represents a partial interpretation of the first K silhouette edges; the leaves of the tree correspond to the M^S complete interpretations of the silhouette. A partially expanded tree is illustrated in Fig. 6-2 for the simplified case where 3 silhouette edges are interpreted in terms of 3 model edges.

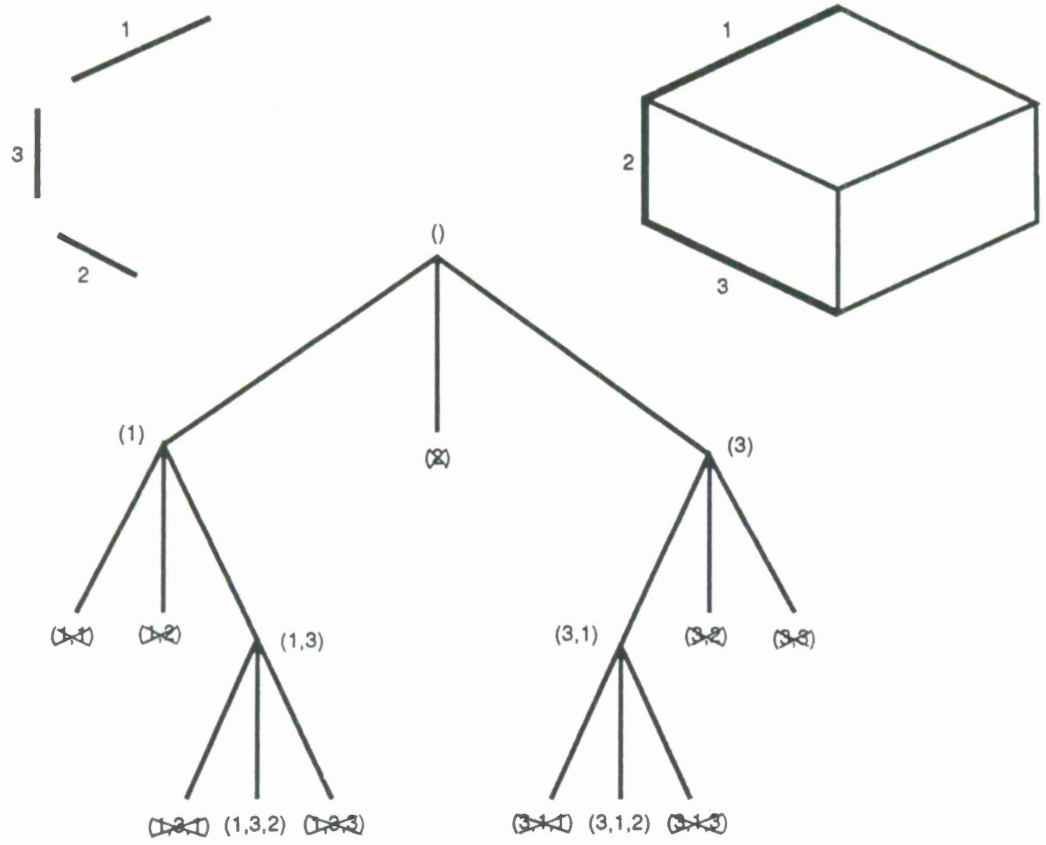


Figure 6-2. Partially expanded interpretation tree for $s = 3$, $m = 3$.

The first level of the interpretation tree corresponds to all possible interpretations for the first silhouette edge. At the second level of the tree, all interpretations for the second silhouette edge are added to the interpretation of the first silhouette edge corresponding to the parent node. For example, the second level node at the extreme left corresponds to the interpretation of both E_1^s and E_2^s as E_1^m. Nodes at the second level are characterized by a list of two labels, $(l_1 l_2)$ where the label l_1 corresponds to the index of the model edge matched to the first silhouette edge, and l_2 is the model index for the second silhouette edge. The tree is further expanded in a similar way, where a node at level K, $(l_1 l_2 \ldots l_{k-1} l_k)$ represents the interpretation of the first K silhouette

edges, which consists of the interpretation of the first $K-1$ edges corresponding to the parent node, $(l_1 l_2 \ldots l_{k-1})$ together with the interpretation l_k for the K^{th} silhouette edge.

6.2 PRUNING THE INTERPRETATION TREE

Once the space of possible interpretations of silhouette edges has been defined in terms of model edges, we require a method by which we can discount large portions of the search space in order to retain only a reasonable number of nodes to actually check for constraint satisfaction. We first describe the process of applying the constraints during the depth-first back-tracking search of the interpretation tree. Next develop suitable constraints that can be applied at each node visited during the search. Finally we examine the modifications to the basic algorithm needed to handle multiple objects in the scene.

6.2.1 Constraint-Based Tree Search

Note that in Fig. 6-2, the second node at the first level corresponds to a match of E_1^s, the first silhouette edge with E_2^m, the second model edge. Since E_1^s is longer than E_2^m, this match can be rejected, as noted on the figure. Note also that the node (1,1) is crossed off on Fig. 6-2 to indicate that this node is also rejected. Indeed, E_1^s and E_2^s are both interpreted as E_1^m, but if they both matched the projection of the same model edge, then they should be colinear, which they are not.

Using simple constraints similar to those mentioned above, it is possible to reject a vast majority of incorrect interpretations. Note that because all the nodes "below" a tree node N contain the partial interpretation of N, a proof that N is an illegitimate interpretation demonstrates that all the nodes below N are illegitimate too. Hence, when a node is rejected, the branch below the node can be conceptually pruned from the tree. This is illustrated in Fig. 6-2 where the rejected nodes have not been expanded.

The test and expansion of the tree may be implemented by any algorithm that will investigate the whole tree. We have chosen a depth-first backtracking algorithm, which has the advantage of requiring almost no storage. The backtracking algorithm only retains information on the current node; the next node to be visited can easily be determined knowing the current node. In brief, the next node is the current node's first child if the node was accepted, or its next sibling if the node was rejected. When the last sibling is rejected, then the algorithm backtracks to the next sibling of the parent node. This tree search strategy is now discussed more specifically, in the case where the current node is $(l_1 l_2 \ldots l_K)$ at level K.

If the current node is accepted, the algorithm next investigates the current node's first "child", $(l_1 l_2 \ldots l_K, 1)$ at level $K+1$. The new node corresponds to the partial interpretation of the current node, augmented by the interpretation of silhouette edge E_L^s as the first model edge E_1^m. If the current node is rejected, the algorithm continues to the next "sibling" of the current node, $(l_1 l_2 \ldots l_k + 1)$ at level K. In other words, when the current label for the K^{th} image edge is invalidated, the next label is tested for that edge. A number of special cases must also be

addressed. If the node was rejected and the label of the last image edge was $l_K = M$, then all the possible choices for the last image edge have been exhausted; therefore, there is no interpretation of the K^{th} silhouette edge which is consistent with the interpretation corresponding to the parent of the current node. This proves that the parent node is not a valid interpretation. The algorithm retracts the parent node and backtracks to the next sibling of the parent, $(l_1 l_2 \ldots l_{K-1} + 1)$ at level $K - 1$. In other words, the label of the penultimate silhouette edge is retracted and the next label is tried for that edge. When a node containing only labels $l_i = M$ has been rejected, then there is no choice for the next node; in other words the tree search has been exhausted.

Finally, in the case where a leaf node ($K = S$) is accepted, the tree search has attained a valid complete interpretation. This interpretation can be stored and/or processed after which the remainder of the tree can be examined by continuing from this node, as if it had tested negatively. In the current system, the verification test is applied to each leaf node before continuing the search. The interleaving of search and verification has the advantage that the verification decision can be exploited in heuristics applied during the rest of the search.

It is clear from the above discussion that the interpretation tree need never be practically instantiated; the only information that the algorithm must retain at any time is the set of labels defining the current node, i.e., the set of labels $(l_1 l_2 \ldots l_K)$. As a consequence, the tree search algorithm itself requires very little storage.

6.2.2 Tree Node Constraint Tests

In the previous section, we indicated two simple tests to reject a candidate interpretation, namely a length test and a colinearity test. Length is the only test that can be applied to individual edge interpretations. A pair of edge interpretations can be tested, in general, for its configuration in the image. In other words, the test checks whether the configuration of the pair of edges in the image is consistent with the configuration of image edges that can be predicted given the 3-D configuration of the matched model edges. Configuration tests for pairs of edges are discussed in detail in Section 4.

It is possible to test configurations of triples and larger sets of edges, but these tests have a number of disadvantages related to the number of tests to perform and the memory required to store threshold values precomputed from the models. Indeed, for tests on T-tuples, the number of tests grows as C_S^T and the memory requirements grow as M^T. These numbers for tests on triples are excessive for typical values of M and T. Therefore, our system only tests constraints on interpretations of individual edges and on interpretations of pairs of edges.

In principle, a node at the K^{th} level of the tree corresponds to the interpretation of K silhouette edges, among which $K(K-1)/2$ different pairs of edges can be tested. However, a node of the tree is tested only after its parent has been verified, so that only the $K - 1$ new pairs of edges introduced by the match of the last silhouette edge must be tested. The number of operations required to test a tree node is proportional to the depth in the tree.

In view of the tree search technique discussed in the previous section, a tree node can be rejected in two different ways. First, a node can be rejected because either a single interpretation

or a pair of interpretations included in the node interpretation violates the constraints based on the matched model edges. This will be referred to as a direct rejection of the node. Second, since the constraints tested on intermediate nodes are only necessary constraints, it is possible for an incorrect partial interpretation to pass the tests. However, the incorrect node will usually be rejected later on by backtracking, for example after realizing that all of its siblings are rejected. This will be referred to as an indirect rejection of the node.

6.2.3 Silhouettes of Multiple Objects

The tree structure and search developed above correspond to a complete interpretation of the silhouette, where each edge of the silhouette must be matched to an edge of the model. When faced with the interpretation of a silhouette corresponding to multiple objects, however, the system may not force a complete interpretation of all silhouette edges in terms of one model; the edges of the silhouette that don't correspond to the model being tested must remain unlabeled. The possibility of leaving an edge unlabeled is also useful when errors in the image analysis create silhouette edges that cannot be matched to the model.

One technique for allowing unlabeled edges is to add to the M model edges an $M+1^{th}$ choice corresponding to "unlabeled" or "null". The first visible consequence of this approach is to increase tree size from M^S to $(M+1)^S$, which may look unimportant. However, a major difference is that "null" labels are always valid so that the $(M+1)^S - M^S$ nodes added to the tree have no inherent constraints; this has disastrous consequences on the tree search. As an example, any valid intermediate node can always be expanded into a valid complete interpretation by adding null labels to the edges that are not yet interpreted. Another consequence is that, although nodes can still be rejected by direct constraints, indirect rejection of nodes becomes almost impossible. When introducing the choice of "null" edges to interpret silhouettes containing more than one object, it is crucial to use heuristics to limit the explosion of the search resulting from the additional unconstrained nodes.

6.3 TREE SEARCH HEURISTICS

In our experiments, we have observed that the tree search is quite effective at investigating the entire search space of edge interpretations with moderate effort. Typically, a search space of 10^{20} interpretations would be exhausted after testing about 10^4 intermediate nodes in the tree. These numbers are quite favorable, but they do correspond to a complete interpretation of the silhouette, where each silhouette edge must be matched to an edge of the model. However, when allowing the "null" label as an interpretation of silhouette edges, the search method described above becomes useless for all practical purposes. It is crucial to develop heuristics to tame the expansion of the tree in the presence of unlabeled edges.

We will describe some heuristics which improve the search in the presence of unlabeled edges and other heuristics which reduce the number of tests both in the presence and absence of unlabeled edges. Among the latter are orderings of edges and backtracking after successful verifications.

Among the latter, we will present constraints on the number of null edges, and a reorganization of the tree.

6.3.1 Ordering of Silhouette and Model Edges

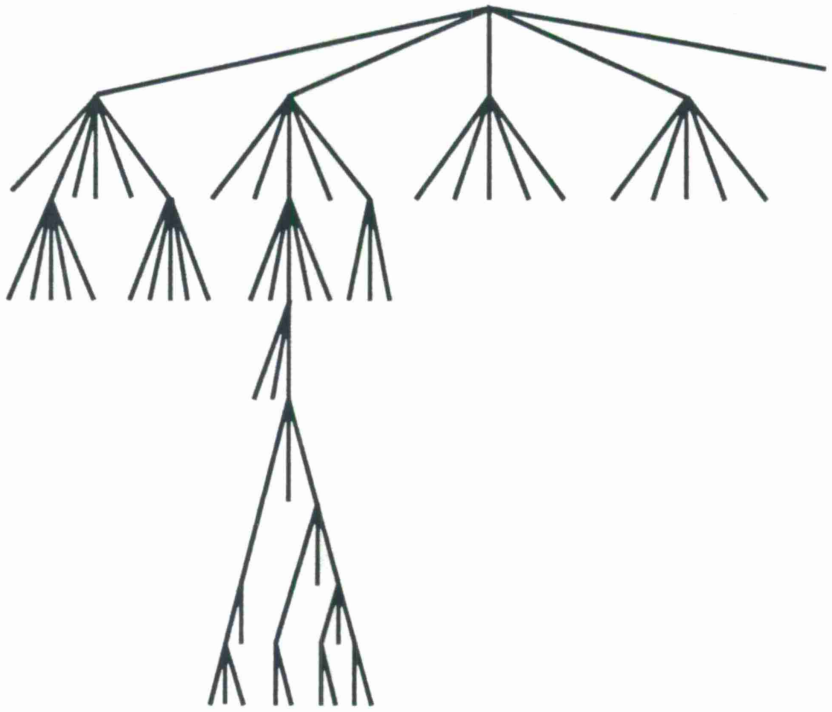
Up to this point, the ordering of both silhouette edges and model edges were considered to be random. It has been observed experimentally that a large fraction of the search is spent on testing intermediate nodes at the lower levels of the tree. These lower levels correspond to the edges positioned in the beginning of the list of silhouette edges. In order to make the search more efficient, we first consider the silhouette edges which are most "distinctive" of its shape. Although "distinctiveness" of edges is difficult to assess, we have observed that the longer silhouette edges usually have better estimates of their position and orientation, and that in addition, constraints on longer edges are more restrictive than those on shorter edges. Therefore, the system orders silhouette edges with respect to their length.

The ordering of edges in the model determines which matches are tried first on the silhouette edges; a desirable ordering of the model edges would first try the model edges that are more likely to match the edges in the silhouette. We have chosen to order the model edges also with respect to their length. Indeed, when testing long silhouette edges, the only model edges likely to match these edges are the longer edges of the model. Since a large fraction of the search effort is spent in the upper levels of the tree, this ordering, which optimizes the search order at those levels, results in very favorable improvements in the search. Typically, an improvement factor of 3 has been observed between random ordering of edges and ordering with respect to length.

In addition to the above advantage, the ordering of model edges with respect to length allows the use of a shortcut in the tree search based on the test of silhouette edge length. During the test of a family of siblings in the tree, corresponding to the M model edge matches for a given silhouette edge, the failure of the length constraint for any sibling rejects all the others. Indeed, if the silhouette edge is found longer than the current model edge being matched, then the node is rejected. Due to the ordering of model edges, all the remaining nodes of the branch correspond to model edges shorter than the current one, so that these nodes will be failed on the length test. Hence, with model edges ordered with respect to length, the failure of the length node can always be followed by direct backtracking. Note that the shortcut developed in this section is only an efficiency improvement of the tree-search algorithm; it does not affect the results of the tree search.

6.3.2 Iterative Verification and Backtracking

In our experiments on the tree search, we have observed that the tree expansion usually has a particular form, which is illustrated by the simplified graph of Fig. 6-3. Basically, the pattern of the tree reflects a relatively large amount of incorrect nodes expanded near the root of the tree, then one or more branches extending to the leaves of the tree, and corresponding to the correct interpretations. There are also some short spurts of incorrect expansions along the branches corresponding to correct interpretations. Oftentimes, a certain number of leaf nodes will be



105340-46

Figure 6-3. Typical expansion pattern of interpretation tree.

accepted at the bottom of the same major interpretation branch. These nodes usually correspond to minor variations on the same correct interpretation. When the constraints on the last part of the silhouette edges are relatively weak, the tree search may accept tens or hundreds of leaf nodes corresponding to the same basic solution. Since all these leaf nodes correspond to almost the same interpretation, they correspond also to very similar views of the object. Large gains in efficiency can be obtained by developing strategies that avoid the verification of each individual interpretation at the leaves of the same major branch.

We have developed an iterative verification scheme which, given a leaf node, will determine the best interpretation along the same major branch of the tree. This algorithm iterates between estimating the viewpoint and refining the interpretation based on the correspondence between the silhouette edges and a synthetic silhouette of the model; this iteration is further detailed in Section 7. After the iterative verification determines the best leaf node along a major branch of the tree, the search may continue past the major branch; this is implemented in our system by allowing the system to backtrack to a level close to the root, after an interpretation is verified. If the iterative verification fails to determine a correct match, then the tree search continues from the initial leaf node.

Note that both the iterative verification and the long backtrack are heuristics which drastically improve the search in some cases, but which remove the guarantee that the algorithm will always find the best match. For example, if the long backtrack is done all the way to the root, the system may miss some symmetrical views of the object. To avoid these misses as much as possible, the system currently backtracks only to the second level of the tree after a successful verification.

Note that the iterative verification alters the tree search path only after a match is successfully verified, so that the search is still guaranteed to find one correct interpretation when it exists.

6.3.3 Limits on Unlabeled Edges

We mentioned previously that the tree search becomes unmanageable when each silhouette edge is allowed to remain unlabeled. This is particularly true because any partial interpretation accepted in the tree can be augmented with null labels for the remaining edges to produce an accepted leaf node. To limit the explosion of the tree due to null edges, the system sets a limit on the number of silhouette edges that can remain unlabeled in a leaf node. Sometimes, it is possible to determine a reasonable figure for this limit from outside information; otherwise, the tree can be searched first with the limit set to 0, then 1, increasing the limit until a verified match is found. These two strategies are discussed in more detail below.

In our discussion of null edges, we determined that the null label prevented intermediate nodes of the tree from being indirectly rejected by backtracking. However, if a limit is set on the number of null labels accepted for the silhouette, then backtracking is possible, although it is slower than in the absence of the null label. We have observed that when the limit on null edges is set to the exact number of spurious edges in the silhouette, the search is quite efficient. However, it is difficult if not impossible to determine this limit from the input data. When the limit is set too low, the correct match cannot be found. When the limit is set too high, the performance of the search degrades in terms of the number of nodes being explored and in terms of the number of spurious leaf nodes being selected; these degradations are usually minor when the limit is too large by only one or two edges. However, the system degrades drastically when the number of null edges is largely overestimated.

To handle the issue of an unknown number of extra edges, the system presents the possibility to search the tree first with the limit on extra edges set successively to 1, 2, ... up to a maximum value, and stops after a complete tree search produces at least one verified match. By default, the system uses this strategy with a maximum of 2 unlabeled edges; this setup correctly handles most identifications of a silhouette corresponding to a single object in the presence of image processing artifacts and special alignments.

Note that the heuristics discussed in this section do improve the search efficiency at the expense of the guarantee to find the correct match when it is present. This compromise is unavoidable in the recognition of silhouettes of multiple objects.

6.4 CONCLUSION

In this section, we have discussed the basic tree search technique as well as a number of refinements for improving its efficiency or generality. In the current implementation, the tree search is interleaved with the verification of the leaf nodes. The verification itself is discussed in the next section.

7. VERIFICATION OF SILHOUETTE INTERPRETATIONS

In the silhouette recognition system, the astronomically huge hypothesis space is efficiently pruned by the tree search technique developed in the previous section. The elimination of incorrect hypothesis is performed by testing simple necessary constraints on the match of each pair of silhouette edges to the corresponding pair of model edges. The necessary constraints guarantee that all the correct interpretations of the silhouette edges are retained, but also leave the possibility that some incorrect interpretations may be retained. In order to remove any incorrect interpretations retained by the tree search, the system performs the verification tests described in this section.

The basic strategy for verifying a candidate interpretation of silhouette edges consists of three steps. First, an imaging transformation is estimated based on the pairings between silhouette edges in the image plane and model edges in 3-D. Second, a silhouette of the model is synthesized for that viewing direction and superimposed on the observed silhouette. Third, these two silhouettes are compared and their differences tested against thresholds to determine if the match is acceptable.

We will first briefly discuss the estimation of the viewpoint given an interpretation of silhouette edges; a more thorough coverage of this complex issue is provided in Appendix B. Second, the synthesis of a model silhouette is discussed, with reference to Section 5. We then discuss the characterization of the differences between the observed silhouette and the silhouette synthesized from the model. Finally, we show how the verification can be refined by an iteration of viewpoint estimation and interpretation update.

7.1 ESTIMATION OF THE IMAGING TRANSFORMATION

The problem of estimating the imaging transformation given correspondences between edges in the 2-D image plane and the corresponding edges in the 3-D model is addressed in this section. Although the problem of estimating an imaging projection from pairs of 2-D and 3-D features has been extensively studied in the case of point features, the problem faced here is substantially more complex since the endpoints of a silhouette edge are not guaranteed to match the endpoints of the corresponding model edge.

Each edge match implies constraints of two different types. First, there is a match between the infinite lines supporting the edges, in other words, the projection of the infinite line supporting the model edge must be superimposed on the infinite line supporting the silhouette edge. The match of infinite lines corresponds to two equality constraints, one for the orientation of the line in the image plane, and one for its lateral position (see Fig. 7-1).

The second type of constraints is related to the longitudinal position of the edges on their infinite support lines. More specifically, the constraint specifies that the silhouette edge must be included in the projection of the model edge. This constraint is best expressed as a pair of inequalities on the longitudinal position of the endpoints of the silhouette edge. Inequality constraints are much more difficult to exploit than equality constraints. Indeed, each individual

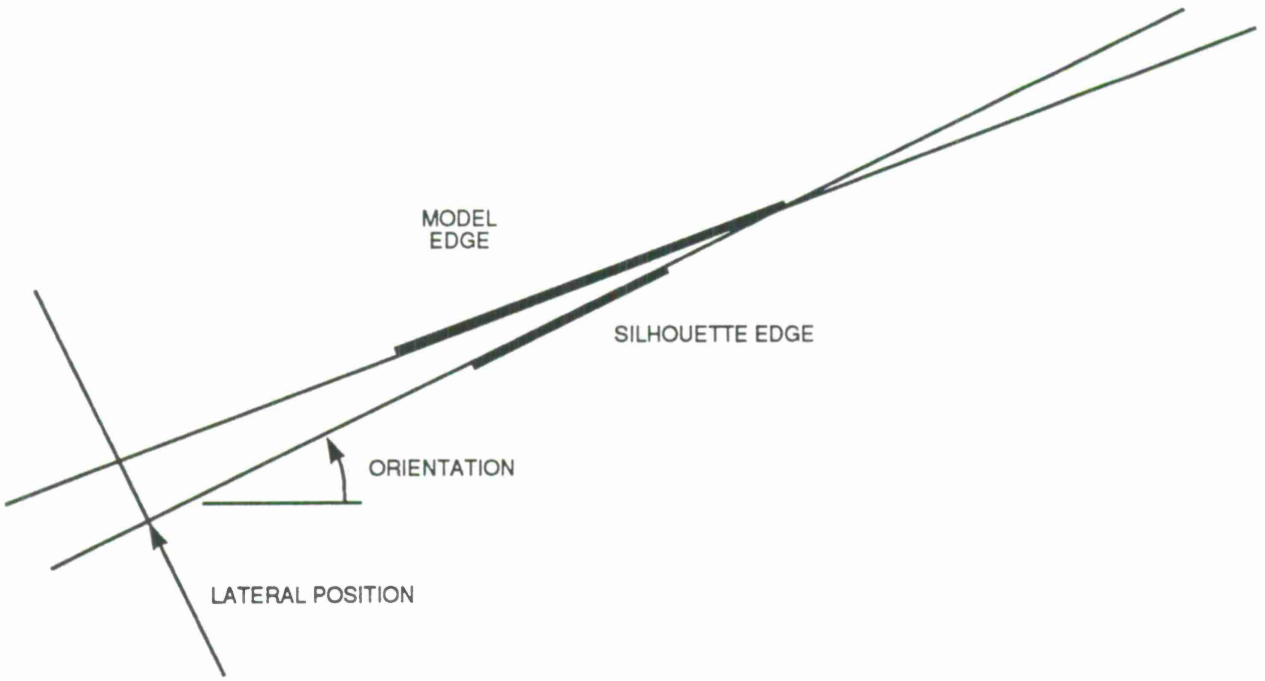


Figure 7-1. Match of a model edge with a silhouette edge.

inequality is either inactive or converted into an equality constraint. In the presence of many inequality constraints, it is very difficult to determine which ones are active and the optimization becomes quite complex. In our system, we have chosen to exploit the equality constraints related to the match of infinite lines only. The inequality constraints linking the positions of the endpoints are not very useful when the endpoints don't match, which is usually the case in our system because of the conservative approach taken for the extraction of straight edges. The imaging transformation is then determined as the one which maps a given set of infinite lines in 3-D to a set of infinite lines in 2-D.

The silhouette recognition system handles pure orthographic projections, or orthographic projections with an unknown scale factor. These transformations have 5 and 6 degrees of freedom respectively. The pairing of one infinite line in the image plane with one in the model provides two constraints on the transformation, namely one for its orientation and one for its lateral position. In principle then, a transformation can be determined, perhaps with a 2- or 4-fold ambiguity, given three independent pairings of image edges and model edges. In practice, the correspondences selected by the tree search generally provide many more constraints than the number of free parameters in the transformation. This redundant information can be exploited to improve the estimate in the presence of noise, for example by finding a least squares solution. Solving for the imaging transformation given pairings of infinite lines is quite complex and is addressed in detail in Appendix B. The authors have not found a closed-form optimal solution for this problem. When the data correspondences provide enough constraints, a suboptimal solution can be determined by first solving for an affine transform with 8 parameters, then finding the orthographic projection that is "closest" to the affine transform. Other solution methods covered in

Appendix B include an exact geometrical method that exploits the correspondences providing five or six constraints only, and an iterative method that converges to the optimal solution when started from a transformation closely approximating the optimum.

7.2 ESTIMATION OF A SYNTHETIC SILHOUETTE

Given an estimate of the imaging transformation relating the 3-D model and the observed silhouettes, a silhouette of the model is synthesized with this transformation so that it can be compared with the observed silhouette. The synthesis of silhouettes of 3-D models was discussed in Section 5. It is worth pointing out that the result of this estimation is a set of straight edges where each one is labeled according to the corresponding edge in the 3-D model (see Fig. 7-2).

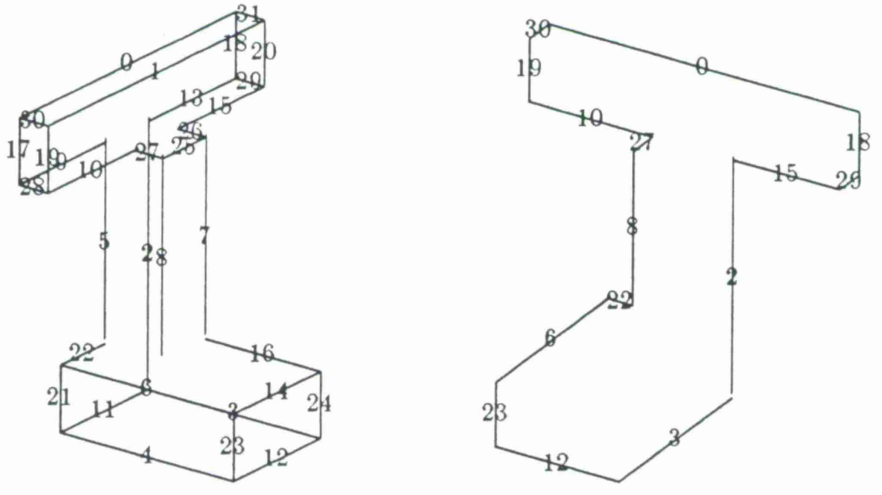
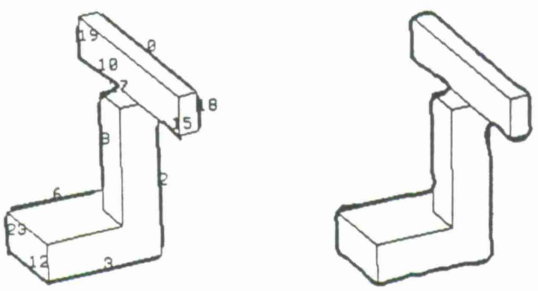


Figure 7-2. 3-D model and a synthetic silhouette with the edge correspondences.

7.3 ESTIMATION OF DIFFERENCES BETWEEN OBSERVED AND SYNTHETIC SILHOUETTES

Given a silhouette extracted from the image and a silhouette synthesized from the model, the system must decide whether to accept or reject the match. This decision is made by first estimating the differences between both silhouettes, then by comparing these differences to thresholds. Note that the image silhouette is represented both in terms of the "raw" silhouette, i.e., the chain of points extracted from the image, and in terms of the image edges extracted from the raw silhouette. Figure 7-3 illustrates this difference. In the figure, the model is represented by its complete image for the sake of clarity; the comparison, however, is based on the silhouette only.

The major purpose of the verification is to test whether the interpretations selected by the tree search are consistent with globally sufficient constraints as opposed to the pairwise necessary constraints that were tested during the search. Therefore, we have chosen to perform the verification



(a) (b)

105340-49

Figure 7-3. Comparison of the synthetic silhouette with the raw silhouette (left), and with the silhouette edges (right).

by comparing the synthetic silhouette with the image silhouette represented by its decomposition into straight edges. On the other hand, the raw silhouette is used after an interpretation is verified, to produce a confidence in the match between the model and the image. These two aspects of the differences between image silhouette and model silhouette are discussed in more detail in the next two sections.

7.3.1 Differences Between Silhouette Edges

The estimation of the difference between the silhouette edges extracted from the image and the silhouette edges synthesized from the model can be decomposed into two steps. First, a difference measure is established between a pair of edges; then, these individual differences are combined to produce global measures for the whole silhouette. In our system, the difference between a silhouette edge and a model edge is characterized by the difference in their orientations, and by the root mean square of the distances between the endpoints of the silhouette edge and the model edge segment. Note that the distance estimate is small when the silhouette edge matches the model edge either as a whole or only in part; the error is large however when the model edge matches only part of the silhouette edge. This property is illustrated in Fig. 7-4(a) and is consistent with our approach to edge matches in the presence of occlusions.

A difficulty arises when the projection of the silhouette model edge is split into two or more parts due to self-occlusions. In that case, the distance is computed separately for each portion of the projected model edge, and the smallest of these is retained; this strategy is illustrated in Fig. 7-4(b). The system does not accept the match of a long silhouette edge with a split model edge because when the model edge is partially self-occluded in the synthetic silhouette, then this occlusion should appear also in the observed silhouette. A particular case of occlusions is the one where the model edge is completely occluded. The error is considered infinite in that case so that the match will always be rejected.

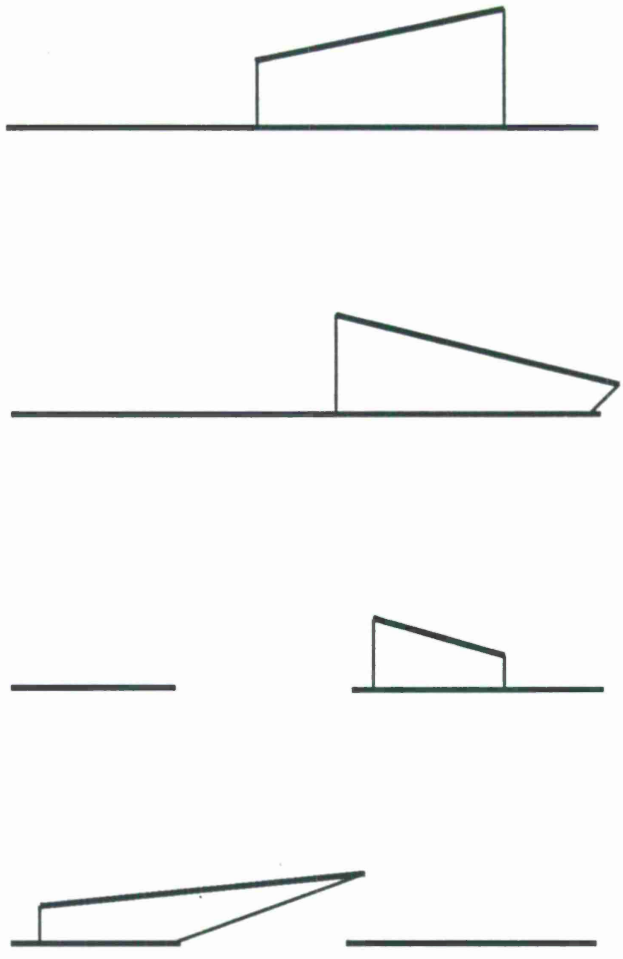


Figure 7-4. Point distances used in the estimation of edge differences; (a)continuous model edge, (b)split model edge.

The above techniques provide two estimates of the difference between a model edge and a silhouette edge, namely one for the orientation and one for the distance. These estimates are combined for all the silhouette edges to produce a maximum orientation error, a maximum distance error, and a global RMS error summarizing the errors over all the edges. In these global measures, the individual errors are weighted with respect to the estimated errors for each silhouette edge. With a perfect estimate of the imaging transformation, the normalized maximum errors is compared to a threshold of 1.0; in other words, a match is rejected if the position or orientation difference between any image edge and the corresponding synthetic edge is larger than the estimated maximum for this error. However, errors in the positions and orientations of the image edges produce errors in the transformation estimated from these edges; these errors may increase the discrepancies between image edges and synthetic edges. In order to account for this additional source of errors, the system uses a default threshold of 1.2 to test the maximum normalized edge thresholds.

In principle, the verification consists of testing a particular leaf of the interpretation tree so that each silhouette edge is tested with respect to the matched model edge only. However, at a higher level of abstraction, the goal of the recognition system is to search for object identities, positions and orientations that interpret the silhouette, irrespective of the identities of matching edges. In that perspective, it seems appropriate to test the observed silhouette with the synthetic silhouette without comparing the edge labels. The difference between both silhouettes is then estimated by comparing each image edge with all the synthetic edges and retaining the figure for the best match. The first technique strictly verifies the particular interpretation of the leaf node in the tree, whereas the second technique corresponds more to verifying the viewing direction estimated from that match. Our system will use either method on demand; this question is further discussed in Section 7.4.

7.3.2 Differences Between Raw and Synthetic Silhouettes

In order to characterize the difference between the image and a synthetic silhouette of the model, it is useful to compare the synthetic silhouette directly with the image silhouette described in terms of individual points. Indeed, the parsed silhouette is useful for determining the object identity, position and orientation of candidate interpretations, but since the edges may not cover all parts of the silhouette, it is useful to return to the silhouette described at the pixel level to estimate a level of confidence in each interpretation.

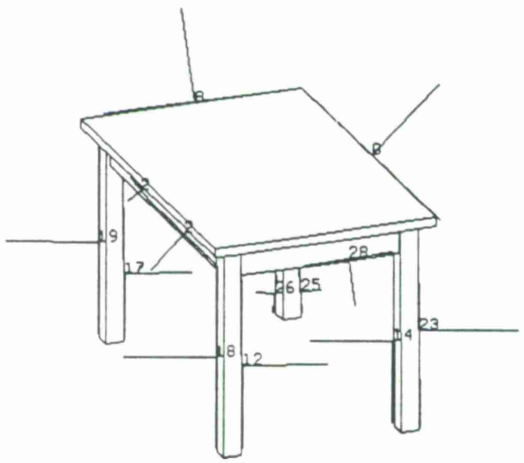
In the case of a silhouette of a single object, the difference is evaluated as the average of the distances between each point of the silhouette chain extracted from the image and the synthetic silhouette. However, when the system interprets silhouettes that may contain more than one object, the distance must be averaged only over the silhouette points which do match the model. For that purpose, the distance estimated for each point of the silhouette chain is first compared to a threshold. Distances below the threshold are averaged as before, and for points with distances above the threshold, the silhouette point is not considered to match the model. The non-matching points are counted to determine in the end, which fraction of the image silhouette matches the model. The match between the silhouette chain and the model is hence characterized by two numbers: (1)an estimate of the average distance between the synthetic silhouette and the matching portions of the silhouette chain, and (2)an estimate of the fraction of the silhouette chain which matches the synthetic silhouette. In our system, the ratio of these two numbers is used as a confidence level for the match.

7.4 ITERATIVE VERIFICATION

We have observed in our experiments with the system that most leaf nodes retained by the tree-pruning search are either correct interpretations or small perturbations on those. In the presence of many small silhouette edges, the number of slightly incorrect interpretations may become large, and the system may have to spend large computation costs on the interpretation of all these. In this section, we develop a method for iteratively improving the interpretation of silhouette edges, when starting from an interpretation close to a correct match. This technique

avoids the exhaustive verification of many similar interpretations by finding the best interpretation with only a handful of iterations.

When the imaging transformation is estimated from an interpretation that closely resembles a correct match, the transformation is usually very close to the correct one so that the synthetic silhouette closely matches the observed silhouette. For example, in Fig. 7-5, the interpretations of the silhouette edges are all correct except for the bottom edge of the short beam of the table in the silhouette, which is interpreted as the bottom edge of the table-top in the model (Edge 11) instead of the actual edge on the beam (Edge 31).

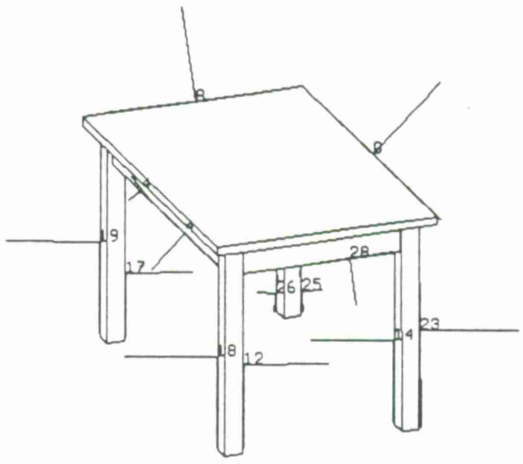


106340-51

Figure 7-5. Silhouette interpretation that closely matches the model.

Because the error is slight and affects only one edge, the estimated transformation closely matches the correct one and the synthetic view of the model closely matches the silhouette edges extracted from the image. Let us consider the estimate of the difference between this edge and the model. If the difference is estimated by comparing the silhouette edge with the projection of the matched model edge in the silhouette, the match will fail because that model edge is occluded and does not appear on the silhouette. However, if the comparison is done with the closest edge in the synthetic silhouette, the match may be accepted, using appropriate thresholds. Beyond the question of acceptance or rejection, the closest edge of the synthetic silhouette also determines the identity of the best matching model edge. This identity may be exploited to improve on the current interpretation of the silhouette edges. In the example above, the system would determine that the synthetic silhouette edge closest to the image edge is Edge 31 and changes the label of the silhouette edge from 11 to 31. The verification can then be tried on this new interpretation, and the process repeated iteratively until a stable interpretation is reached. The resulting match, illustrated in Fig. 7-6, provides a much closer fit between the model and silhouette.

Clearly, the updated match discovered by the iterative verification is a leaf of the interpretation tree and since this leaf corresponds to a consistent interpretation, it must be retained in the tree search. Therefore, if the tree is exhaustively traversed, this node will be reached at another point in the search. However, we mentioned in Section 6 that the tree search may sometimes



105340-52

Figure 7-6. Updated silhouette interpretation.

retain many leaf nodes where the interpretation differs by one or a few edges only. With the classical verification technique, many of these interpretations may be verified, although with varying degrees of confidence. The best interpretation is found but with a potentially large computational cost. On the other hand, the iterative verification finds the best interpretation with a limited effort. After this interpretation is found, the search can backtrack high up in the tree.

Additional issues appear in the iterative verification when the system accepts null interpretations for the silhouette edges. Indeed, the update may then either change the label on an edge, turn a labeled edge to unlabeled, in other words, discarding it from the match, or turn an unlabeled edge into a labeled one, thereby incorporating it into the match. These possibilities enhance the power of the iterative verification, but also increase the probability of problems such as oscillations between two interpretations or divergence towards an interpretation that excludes all the edges. These problems have been largely eliminated in our system by setting several different thresholds for the various decisions. First, the edge errors are always tested against a relatively high threshold, to quit the iteration if the match is blatantly false. Second, for switching an edge between labeled and unlabeled, the error of an unlabeled edge is tested against a low error threshold to decide on whether to label it, and a higher threshold is used for adopting the null label on an edge that was previously labeled. The hysteresis in these transition levels between labels and nulls prevents oscillations. Finally, a lower threshold value is used to test the final interpretation. In addition to the error thresholds, a threshold is set on the number of edges that may remain unlabeled. Good results have been observed with this threshold slightly higher than the limit used during the tree search.

Figure 7-7 illustrates the mechanism where the interpretation of an edge may change from labeled to unlabeled or vice-versa. In the example, the model of a table in (a) is matched against the silhouette edges in (b). Note that these edges correspond to both the table and another object; this example is further discussed in the next section. We are interested here in the iterative update

of the edge labels. The edge labels first selected by the tree search are displayed in (c). Although many edges in this interpretation are correctly labeled, some labels correspond to different model edges in the same proximity. Because of these minor discrepancies, the synthetic image of the model does not fit the silhouette edges exactly. In particular, the silhouette edge labeled 11 is quite far from the closest synthetic silhouette edge. In the next iteration, shown in (d), this particular edge is taken out of the match and the edge label 3 is updated to the correct value 7. As a result of these label updates, the synthetic image of the model for the new interpretation fits very closely with the silhouette edges. The edge formerly rejected is incorporated back into the match as Edge 31, as shown in (e), thereby providing additional support for the match. This example justifies the use of two thresholds in the test of the synthetic silhouette and the observed silhouette. Indeed, the initial match is not satisfactory by itself, but its relatively low discrepancy suggests that a better match may be found by the iteration.

7.5 SUMMARY

In this section, we have discussed the verification of the interpretations retained by the tree search by estimating an imaging transformation, synthesizing a silhouette of the model for this transformation, and finally comparing the silhouettes extracted from the image and synthesized from the model. The comparison provides an acceptance/rejection decision for the interpretation. In addition, the comparison can be exploited to iteratively improve the interpretation, potentially avoiding the costs of multiple verifications on tree leaves corresponding to closely related interpretations.

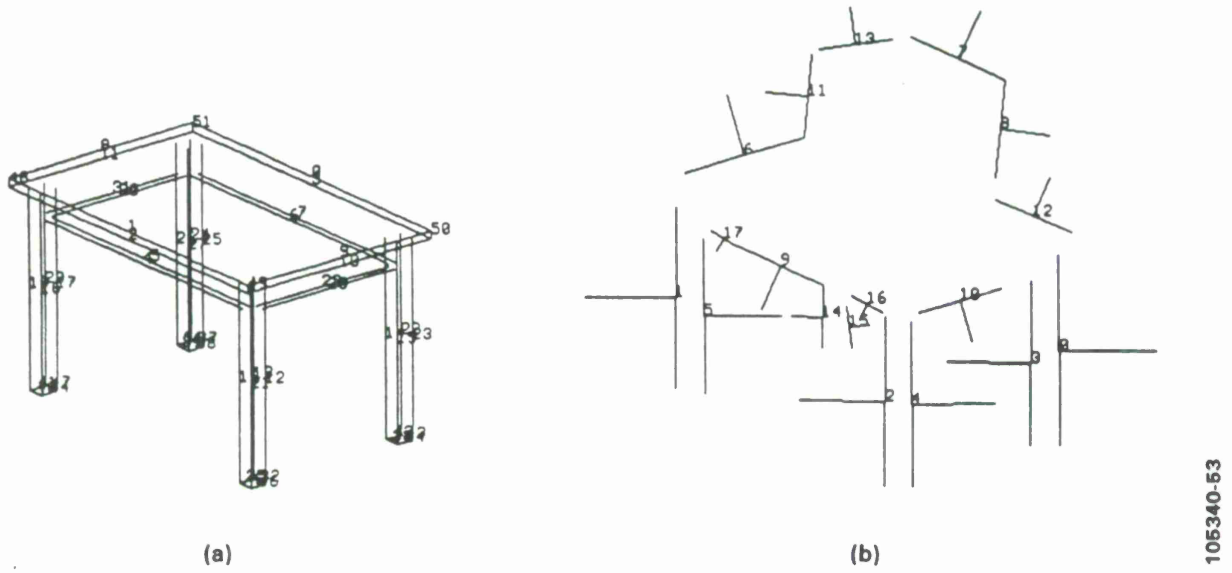
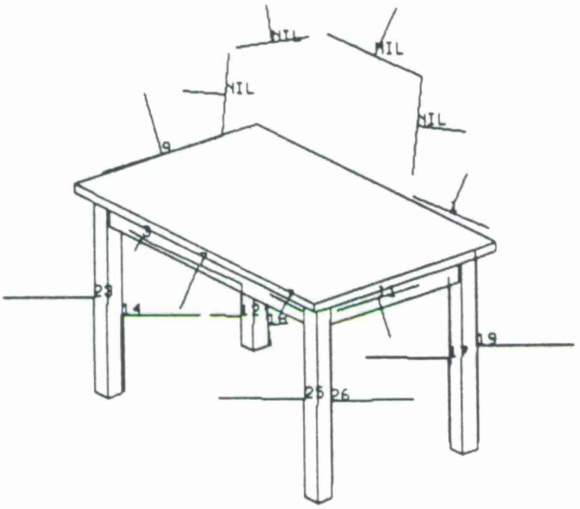
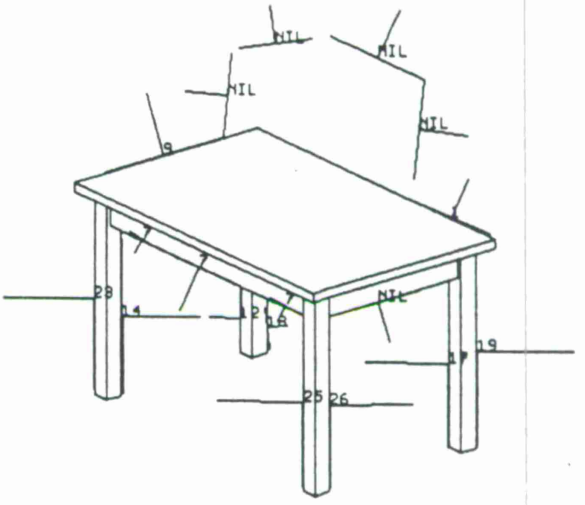


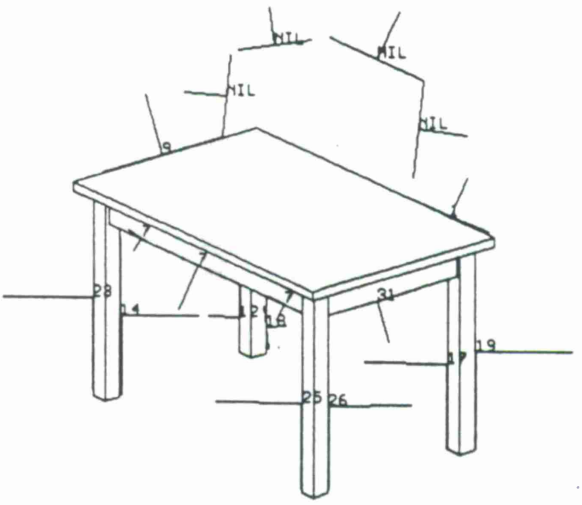
Figure 7-7. Iterative update of edge interpretations.



(c)



(d)



(e)

105340-54

Figure 7-7 continued.

8. EXPERIMENTAL RESULTS

In this section, a number of examples illustrating the performance of the SILC system are presented. Tests on individual examples discussed in Section 8.1 demonstrate the system performance in a number of situations of interest. Several series of experiments have also been performed to better characterize the system performance, in particular its variation with various settings of the system parameters. These experiment series are discussed in Section 8.2. In both the simple experiments and the experiment series, both the accuracy of the decisions and the computational costs are discussed. In Section 8.3, a few issues encountered with the system are discussed.

To this time, the system has been tested on synthetic images only, in part because of a lack of appropriate imagery, and because of the difficulty of obtaining accurate models for the objects present in available images. However, the results presented here are indicative of the basic performance of the algorithm, including its response to noisy data.

8.1 SIMPLE EXAMPLES

Here we present a number of examples to demonstrate the system performance in a number of contexts. The first example illustrates the system response in the recognition of a simple polyhedral object in a moderately high resolution silhouette image. Two performance measures are then described and illustrated on the results of the first example. The first measures recognition accuracy, and the second measures computational cost of recognition. Subsequent examples show recognition in the presence of noise, curved surfaces, occlusions, and multiple objects.

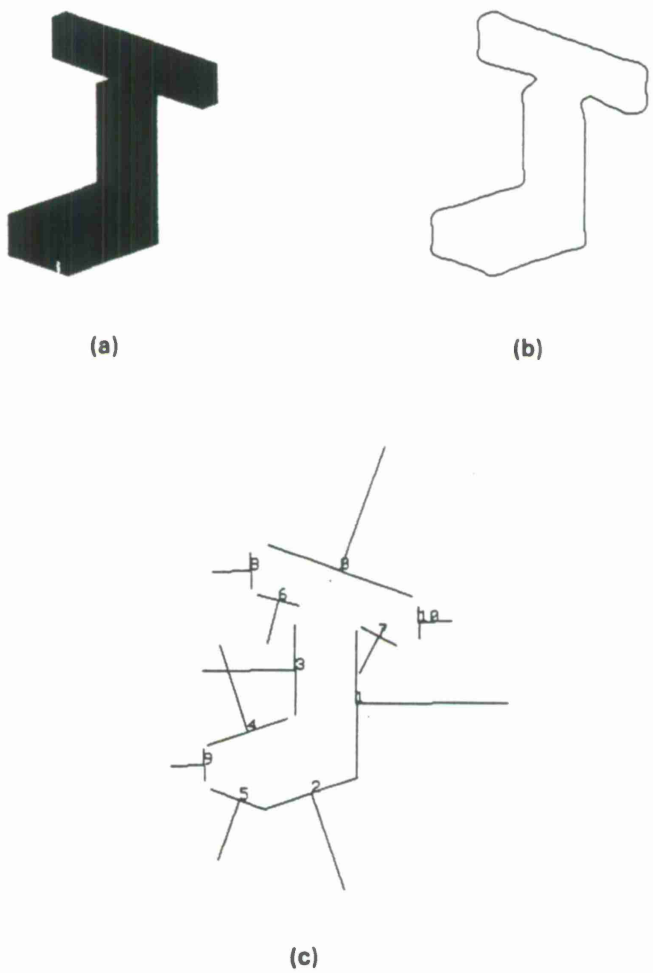
8.1.1 Simple Polyhedral Object

Figure 8-1 illustrates the match of the silhouette of a simple polyhedral object with the model for that object. The solid silhouette in (a) was synthesized from the model, then subsampled by a factor of 4 so that the resulting silhouette was about 40 pixels across. The image was then convolved by the Laplacian of a Gaussian filter and the zero-crossings detected to produce the chain in (b). The silhouette was parsed by the Ramer polygonal approximation method. After the shorter edges of the approximation were removed, the result in (c) was obtained. The configuration of edges in (c) is then compared with the 3-D edges of the model by the tree-search method. The labels in (d) were retained by the search; a synthetic view of the model for the viewpoint estimated from the interpretation is superimposed on the labels in (d). The comparison between the observed silhouette edges and the synthetic silhouette revealed that the match is acceptable, as shown in (e). Finally, the original silhouette chain is compared to the synthetic silhouette as illustrated in (f). The match was given a high confidence figure of 5.34. The noise parameters were set according to the levels due to discretization noise in the picture; the scale uncertainty was set to 10%.

The same silhouette was also compared to a different object to show the correct rejection of false matches. The search for a match with the model displayed in Fig. 8-2 rejected the hypothesis while allowing 3 unmatched silhouette edges.

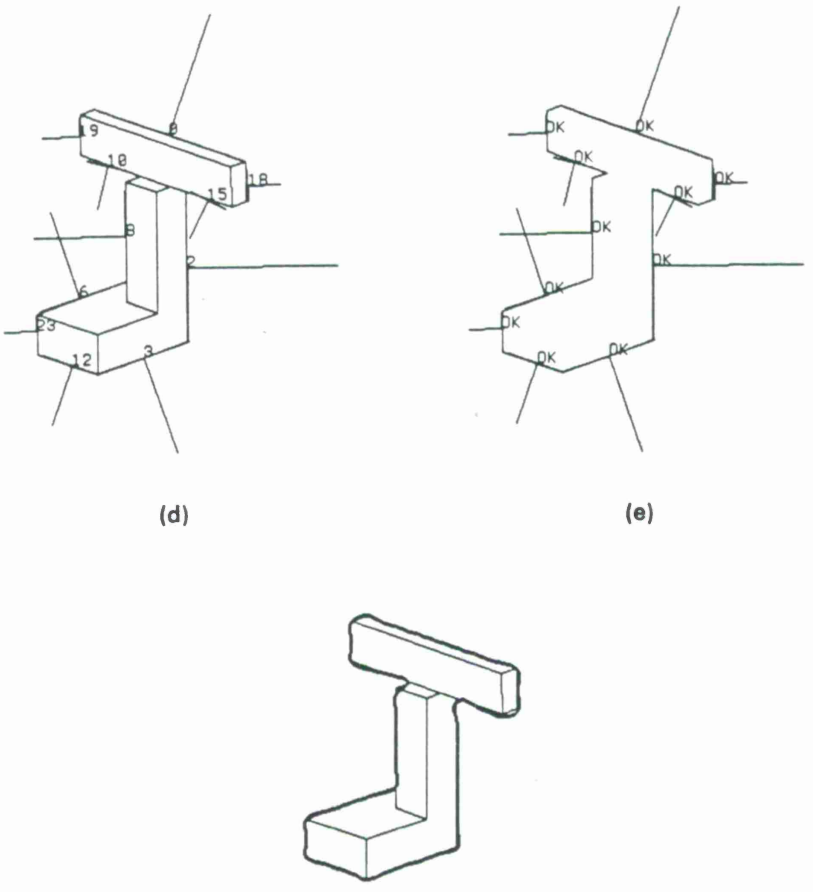
8.1.2 Performance Measures

Performance measures for recognition accuracy and cost are presented in this section. Their application to the first example is described in detail. These measures will be used in the subsequent examples to provide a relative measure of increasing complexity as well as an absolute measure of the system performance.



106340-55

Figure 8-1. Recognition of a simple polyhedral object.

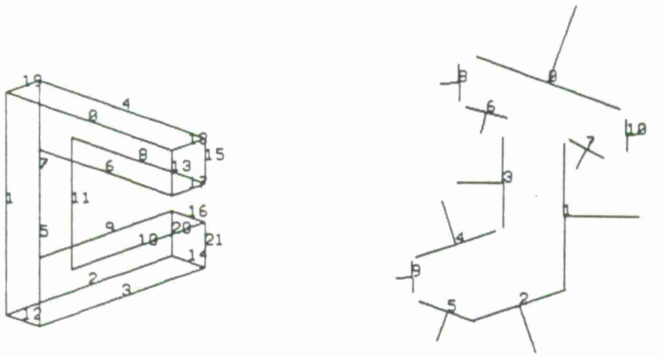


105340-56

Figure 8-1 continued.

Recognition Accuracy

In the example in Fig. 8-1, only the correct interpretation of the data was retained by the system. The test with other models resulted in those matches being rejected. This example shows a perfect record. In general, more than one interpretation of the silhouette in terms of the object models may be retained. These ambiguities are natural to the problem when those interpretations are justified by the tolerances given to the system. The SILC system produces simple confidence factors to address this problem. To test the similarity between edge chains and a synthetic silhouette of the model, the distance between each point of the chain and the synthetic silhouette is evaluated. When this distance is lower than a threshold error for the silhouette, the point is considered matched; the error is incorporated in a mean square error for the matched points. When the distance is higher than the threshold, the point is counted as unmatched. Two measures are estimated by this procedure: the fraction f of silhouette points which are matched and the averaged error e between matched points and the synthetic silhouette. In SILC, the



105340-57

Figure 8-2. Test of silhouette edges with a different model.

confidence factor of a match is considered as the ratio f/e. A dimensionless quantity is obtained by normalizing the error e by the estimated error for the type of imagery. A factor of 1 would be obtained for a model matching 100% of the silhouette with all the silhouette points at the maximum error distances. Values larger than 1 usually correspond to interesting matches.

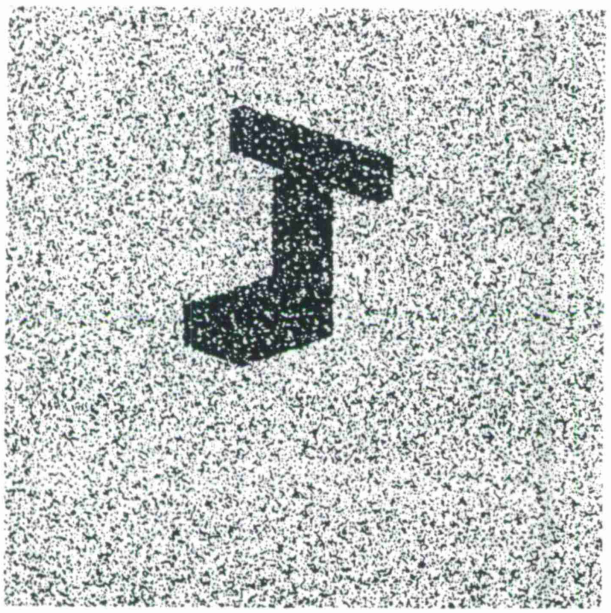
Computational Cost

The computational effort required for the match in Fig. 8-1 is divided in the computation of the silhouette tables, the tree search, and the verification. The tree search explored only 270 nodes in this example (~50 ms); only one synthetic silhouette was compared with the image silhouette (~20 ms). The recognition of the object is hence extremely fast. The search for a match with the model displayed in Fig. 8-2 rejected the hypothesis after searching 266 nodes when no null edges were allowed. Searching for matches with increasing numbers of null edges from 0 to 3 required 27,604 node tests (1.38 s) to reject the hypothesis. The search increase with null edges is discussed later. This simple example shows that the search is extremely efficient for simple problems.

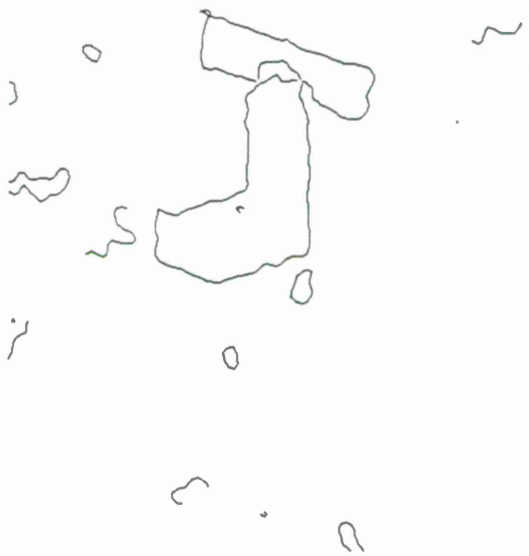
8.1.3 Image Noise

The example in Fig. 8-3 illustrates the performance of the system with a moderately noisy image. The silhouette image in (a) was obtained by flipping at random, 30% of the points in a binary image of the object. After subsampling by 4, convolving with a Laplacian of Gaussian and detecting the zero-crossings, the edge chains in (b) were retained. These correspond to all the edge chains with a median contrast of at least 60% of the chain with the highest median contrast. Note that the silhouette of the object of interest is separated in two pieces and that spurious chains are retained by the system. After parsing the silhouette with the Ramer polygonal approximation and after removing the shorter edges, the silhouette edges in (c) are obtained. The match is attempted with the correct model in (d). The tree for this example has about 10^{30} nodes. The tree was pruned while allowing 10 unmatched edges and 10% scale difference between silhouette

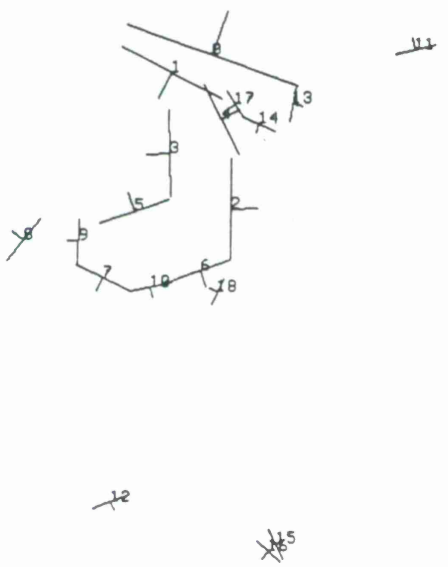
and model. Less than 50,000 nodes were tested in this tree to retain 2 potential matches, shown in (e). These two interpretations were both accepted by the verification subsystem since the quasi-



(a)



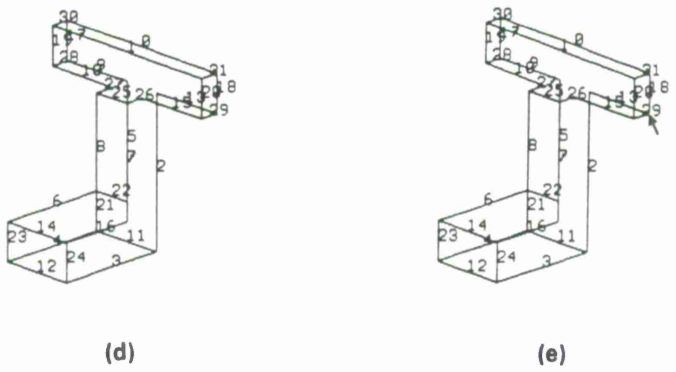
(b)



(c)

105340-58

Figure 8-3. Recognition in the presence of image noise.



(d) (e)

105340-59

Figure 8-3 continued.

symmetry of the object and the large edge errors make this second interpretation consistent. However, the confidence figures, 1.44 and 0.30, are greatly in favor of the correct interpretation.

8.1.4 Object With Curved Surfaces

The example in Fig. 8-4 illustrates the recognition of an object with curved surfaces approximated by a polyhedron. The binary image in (a) was processed with the Laplacian of Gaussian edge detector to obtain the silhouette in (b). This silhouette contains both long straight edges and curved parts. The parsing of the silhouette is performed by first extracting the long straight edges, then modeling the remaining smooth portions of the silhouette by individual points. This method of silhouette parsing is discussed in detail in Appendix A. The result of this parsing is displayed in (c). Note that the two curve segments are modeled by strings of zero-length edges. These zero-length edges can be matched with a polyhedral model of the object (d) without the problems associated with matching a polygonal model to an observed curve modeled by a polygon. The tree was searched with numbers of null edges increased from 0. The scale tolerance was 10% for this example. The interpretation in (e) was selected by the search and the verified when superimposed with a synthetic image of the object, as shown on the figure. The system also selected and verified the symmetric interpretation of the silhouette, see (f). The search was completed without introducing null edges; fewer than 23,000 nodes were tested in a tree containing approximately 10^{26} nodes. A total of 4 leaf nodes were selected; 5 verification tests retained the two interpretations in (f). Due to the symmetry of the problem, these interpretations have exactly the same confidence of 4.57.

8.1.5 Occlusions

The example of Fig. 8-5 illustrates the recognition of an object, a table, given only a partial silhouette. The silhouette edges in (b) extracted from the binary image in (a) cover only part of the silhouette. The tree of interpretations of these edges in terms of those of the model in (c)

was run with a scale tolerance of 10%. The tree has about 10^{15} nodes; the search tested 5600 nodes and found no match with no null edges, since Edge 1 in (b) clearly doesn't match the model. With one null edge, the search tested 20,400 nodes and selected 12 edge interpretations. The four interpretations in (d) were verified; these correspond to symmetric views of the object. All these views were assigned the same confidence. The synthetic model silhouette accounts for 75% of the observed silhouette chain in all 4 cases. When compared to the interpretation of a complete silhouette, the interpretation tree is smaller when fewer edges are present; as a

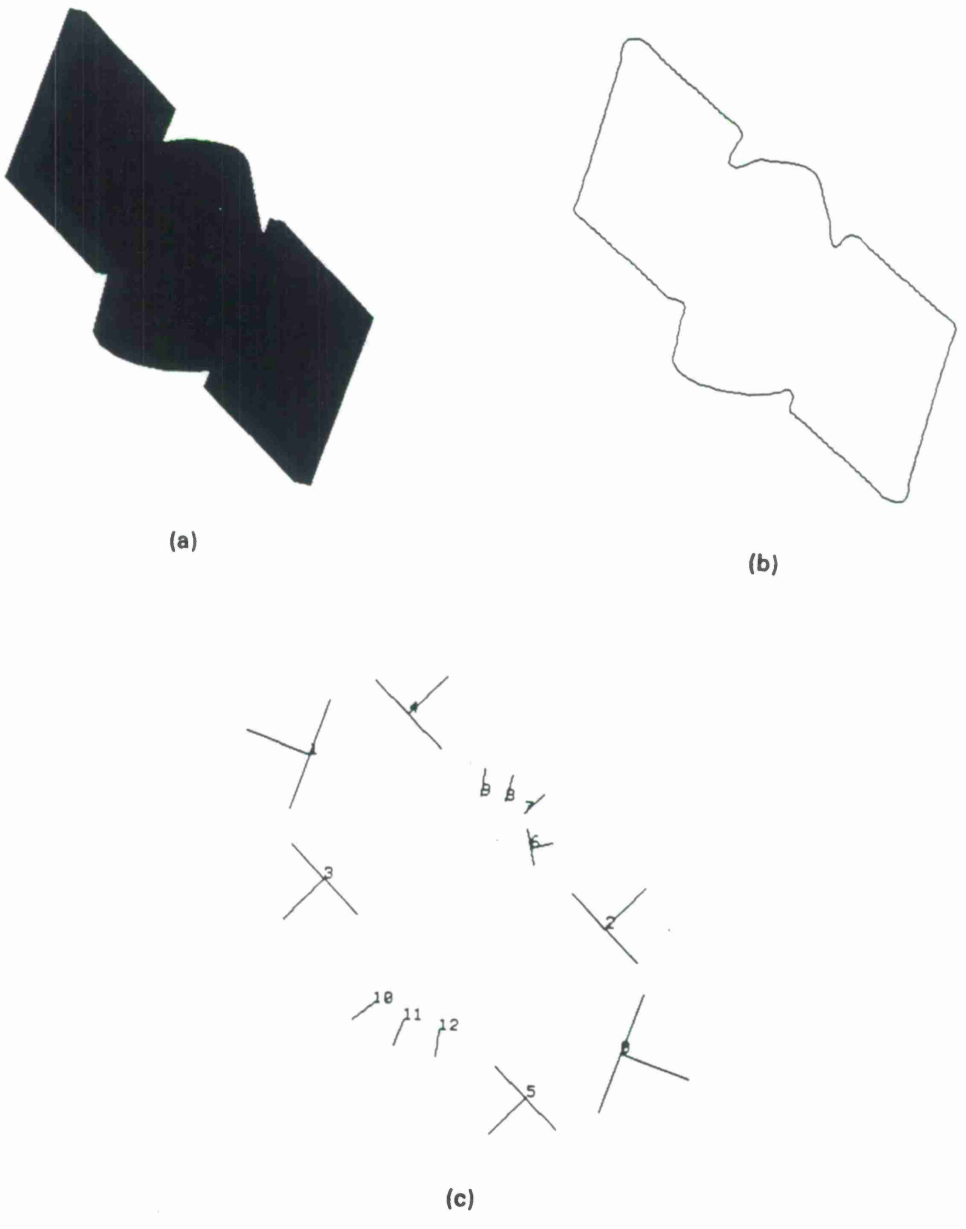


Figure 8-4. Recognition of an object with curved surfaces.

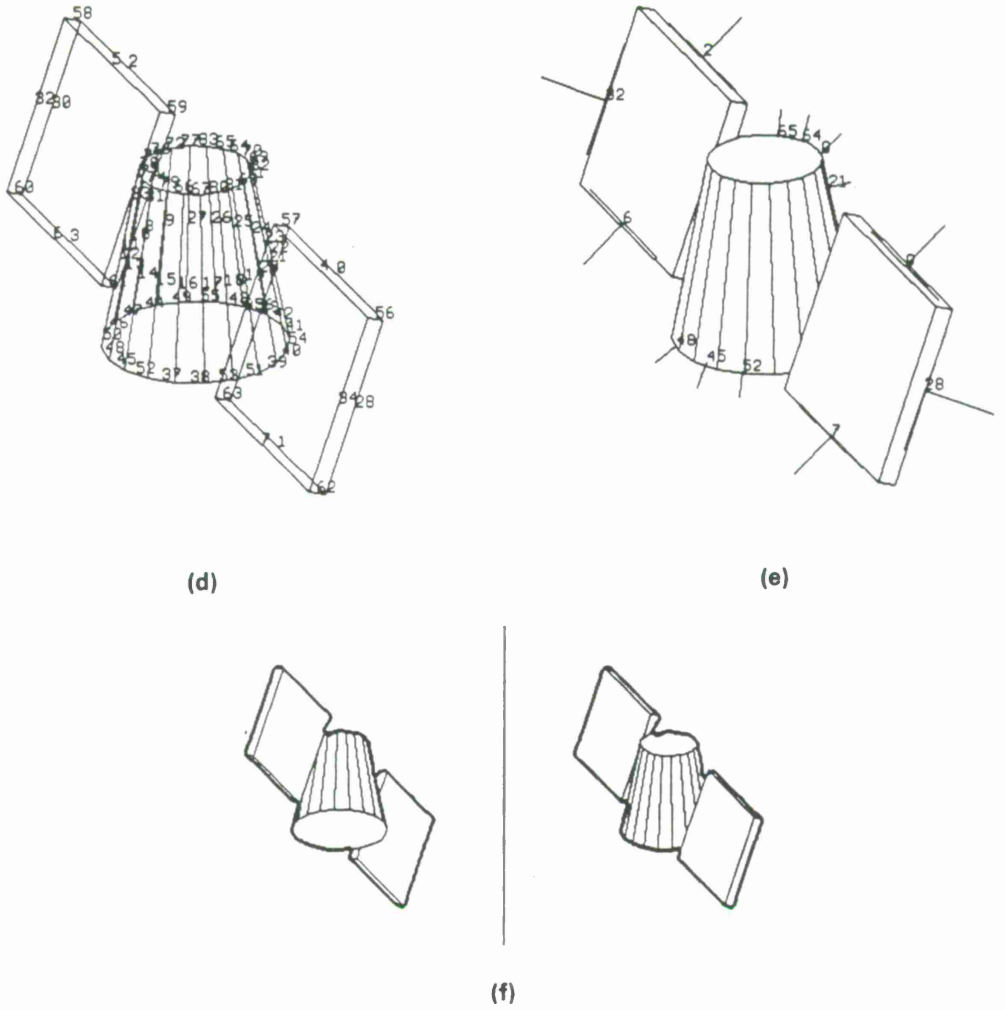


Figure 8-4 continued.

consequence, the tree search will need to test fewer edges. However, the constraints are weaker with a partial silhouette since fewer edges need to be interpreted. As a result, the search is more likely to retain incorrect interpretations which the verification must later reject. Because of the high computational cost of verification compared to tree node tests, the cost of recognition will usually be higher for a partial silhouette.

8.1.6 Multiple Object Silhouette

Figure 8-6 illustrates the recognition of an object from the silhouette in (a) which corresponds to two objects [see the scene in (b)]. The system attempted to find an instance of the table model in (c) among the silhouette edges in (d), extracted from the binary image in (b). The interpretation tree was searched with increasing numbers of null edges. No matches were found with 0,

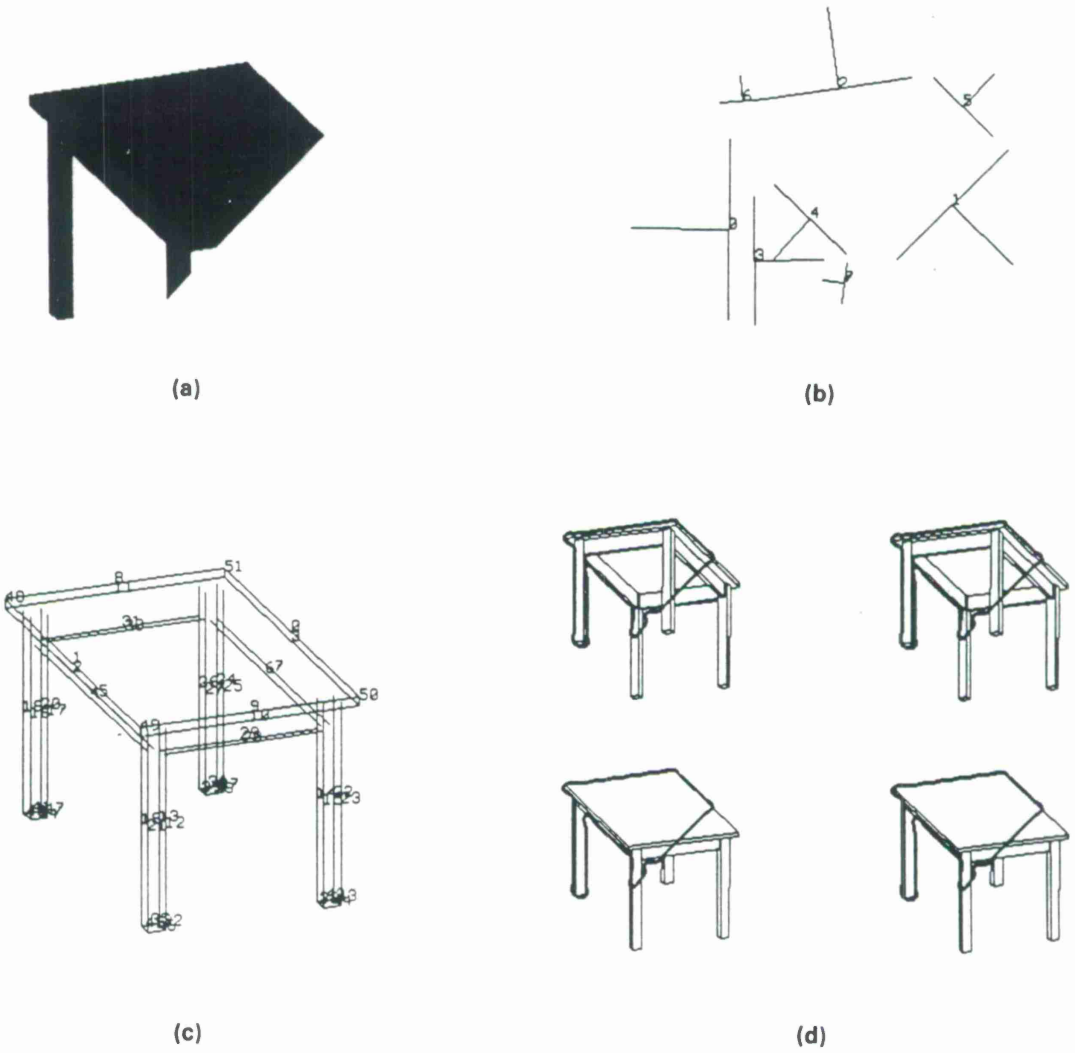


Figure 8-5. Recognition of an object with curved surfaces.

1, 2, and 3 null edges; the search required respectively 5,400, 29,300, 81,700 and 174,000 node tests to reject those possibilities. The search with 4 null edges found 4 acceptable interpretations of the data which were all verified; one of these is shown in (e), after iterative verification. An additional 102,000 nodes were tested to search the tree with 4 null edges. These four interpretations correspond to the symmetry of the model; their comparison with the raw silhouette chain is illustrated in (f).

It is interesting to note that the unsuccessful search with 3 null edges required more node tests than the successful search with 4 null edges, although the full tree was searched in both cases. This is mainly due to the heuristic allowing a backtrack high in the tree after a successful verification. In total, the search for interpretations of the silhouette edges in (d) required about 392,000 node tests. Note that if the number of null edges were exactly known, only 102,000 would be necessary. In practice, it is rare that the number of null edges can be estimated exactly.

Instead of trying the numbers of null edges from 0 on, a different strategy is to pick an upper bound on the number of null edges and to perform the match with that number of nulls. In the example shown, the search with 8 null edges required 296,407 node tests, fewer tests than when the number of nulls is started from 0. However, this approach may retain more incorrect matches due to the higher tolerance of the search. Indeed, the search with 8 nulls retained 18 interpretations, and the verification confirmed 8 of them. The 4 interpretations derived by the other method were among the 8 verified with 8 nulls; they were given higher confidence measures that the other 4 verified matches.

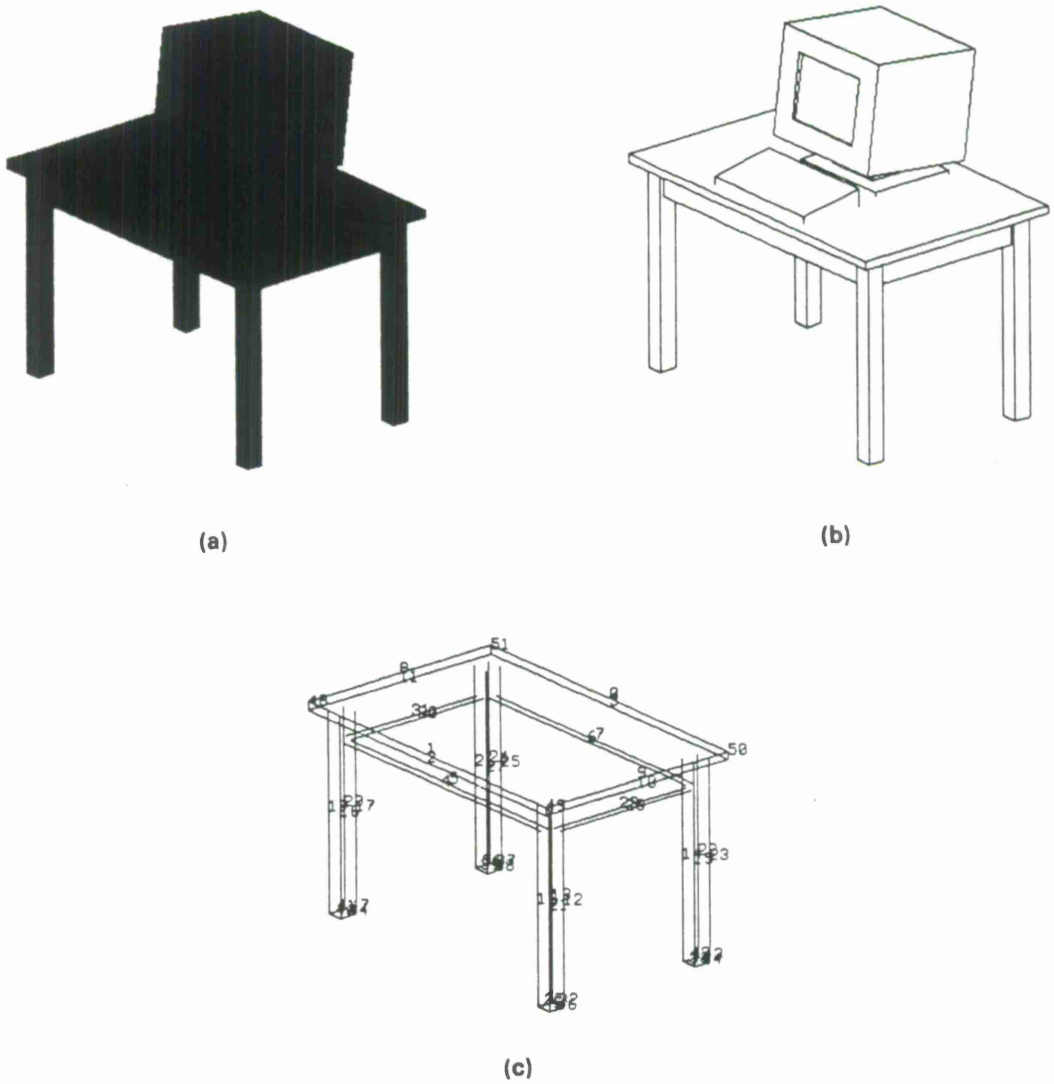


Figure 8-6. Recognition from a multiple object silhouette.

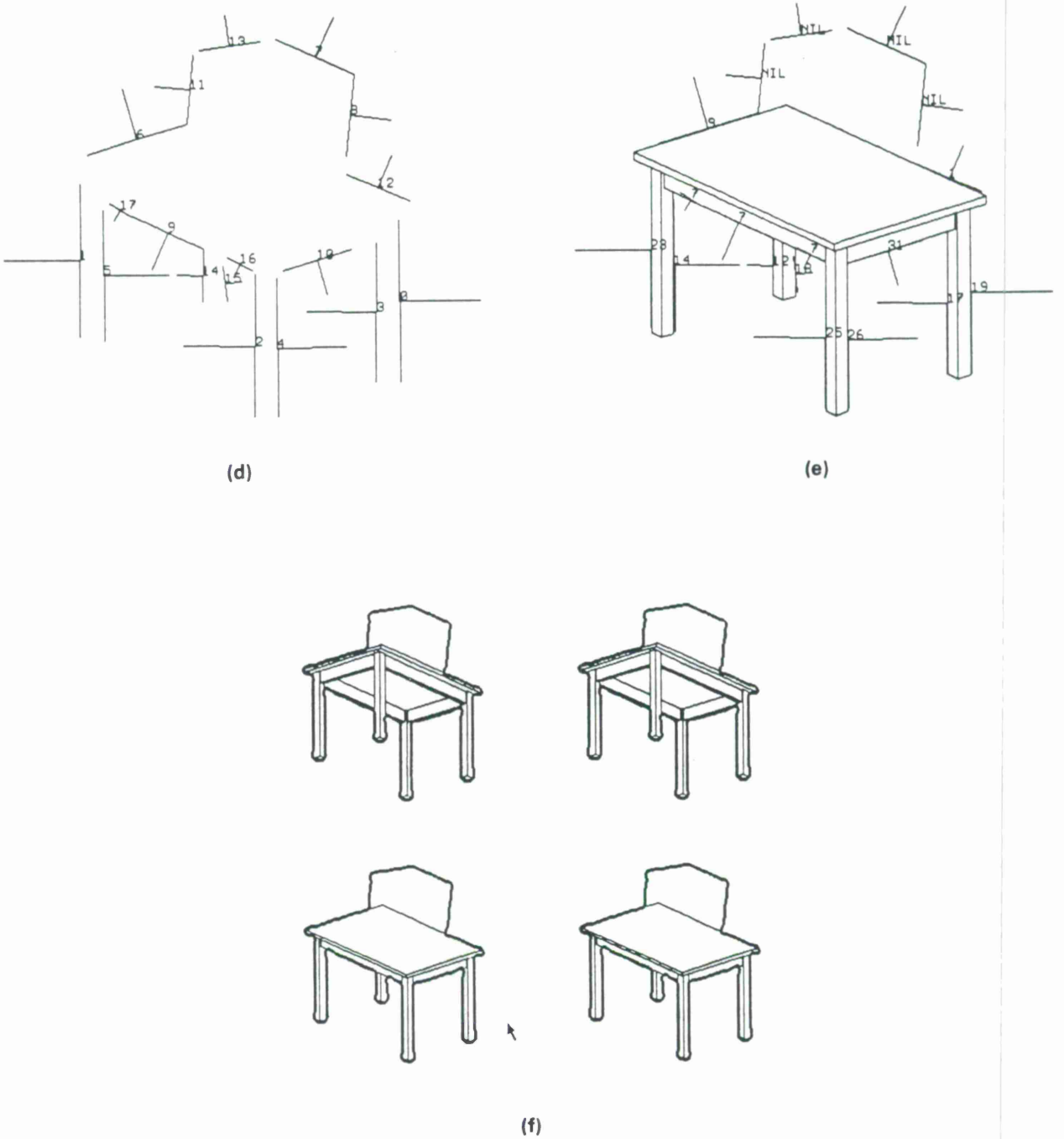


Figure 8-6 continued.

8.1.7 Summary

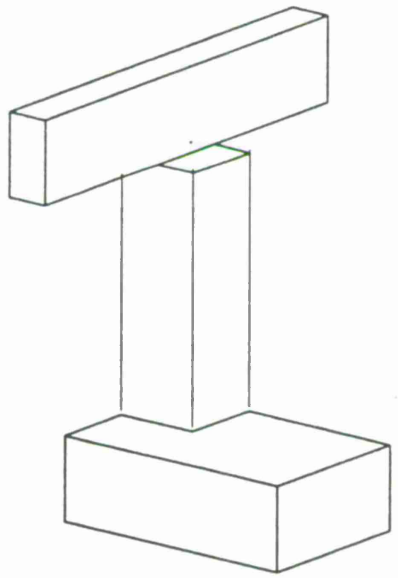
In this section, a number of experiments were presented to demonstrate the performance of SILC in the face of noise, scale uncertainties, partial data, and extraneous data. As long as

these degradations do not exceed their estimated bounds, the system will retain the correct interpretation of the data. Note that when one particular degradation is slightly larger than its estimated bound, this effect may be accounted for in terms of another degradation. For example, if the difference in scale between the model and the image is 12% while it was estimated below 10%, the extra 2% may be accounted for as noise.

When the bounds become looser, additional interpretations of the data may become possible so that the system will spend more effort in searching and verifying these. It is interesting to determine the increase in search and verification effort on one hand, and on the number of false interpretations on the other hand, when bounds on the input data degradations are loosened. An assessment of these trends is developed in the next section.

8.2 SERIES OF EXPERIMENTS

In this section, results on moderate numbers of experiments are reported. These experiments were used to characterize trends in the system performance as the tolerance to input degradations is increased. These experiments demonstrate that the performance degrades when the tolerance is increased, but that this degradation is graceful. The degradations considered here include discretization noise, scaling, and spurious edges.

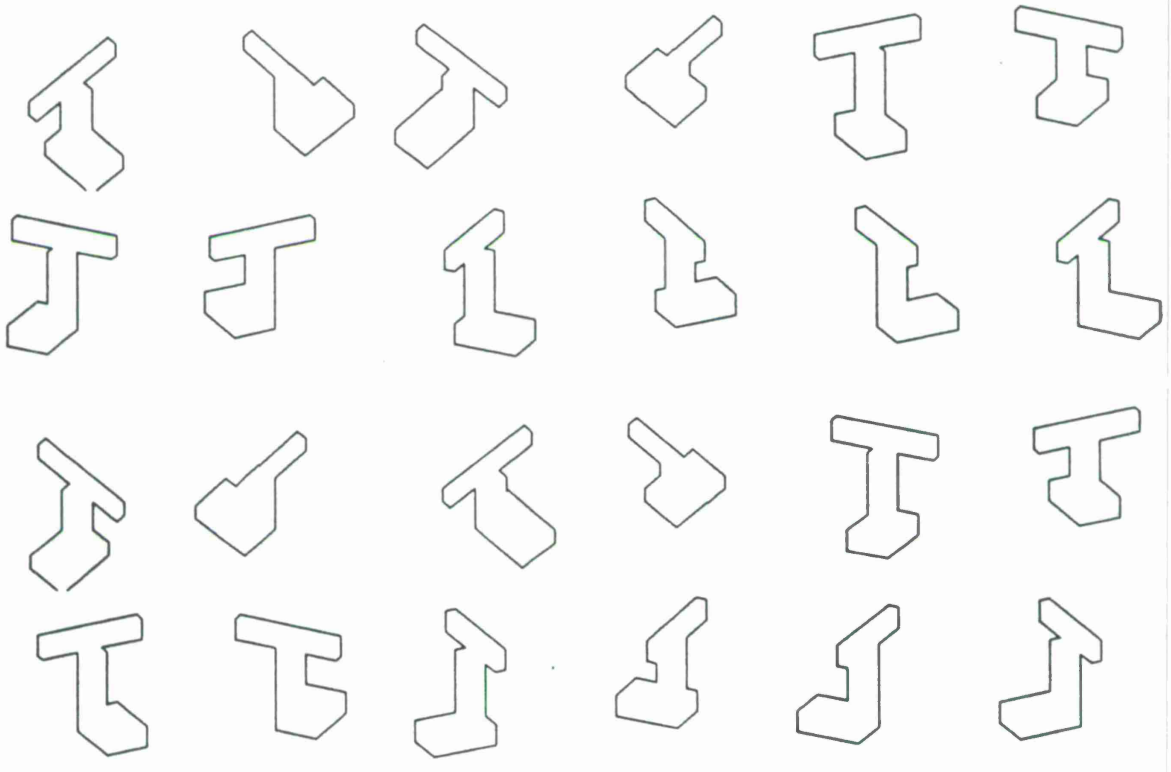


105340-65

Figure 8-7. Model for recognition series.

The tests presented in this section relate to the match of the model in Fig. 8-7 with synthetic silhouettes of this object taken from 24 different viewpoints, (see Fig. 8-8). The cost of this match is estimated by number of tree node tests and the number of verification tests between a synthetic silhouette and the "image silhouette." These estimates can then be converted to execution times. The execution time of a node test depends on its depth in the tree and the time for a verification

depends on the size of the silhouette and of the model. With moderate size examples, good estimates of recognition times were obtained by counting 200 microseconds for a node test and 20 ms for a verification. The results are reduced to extrema and medians over the 24 viewpoints, and compared for various settings of the parameters.



105340-66

Figure 8-8. Compiled views of the model.

The first test concerns the performance of the system when the tolerance with respect to scale is increased. Figure 8-9 illustrates the increase in the numbers of node tests and in the number of verification tests with the scale tolerance. The graphs show that the increase is gradual and that the computational costs are reasonable for tolerances up to 50%. For the higher scale tolerances, the system verified more than one interpretation for some views.

The second series of experiments tests the system performance in the presence of noise. The results in Fig. 8-10 illustrate performance variations in function of quantization noise, when the same original images are subsampled by various factors, thereby affecting the target size measured in pixels. In these examples, degradations of the system performance are relatively severe for the lower discretization; however these correspond to quite extreme cases where the silhouette is only 20 pixels across in the image. The results in Fig. 8-11 show the dependence of computation times on estimates of the amount of noise in the image. When the edge locations are estimated with subpixel accuracy, which is usually the case, the increase of computational costs is smooth. For large amounts of noise however, the added tolerance largely increases the

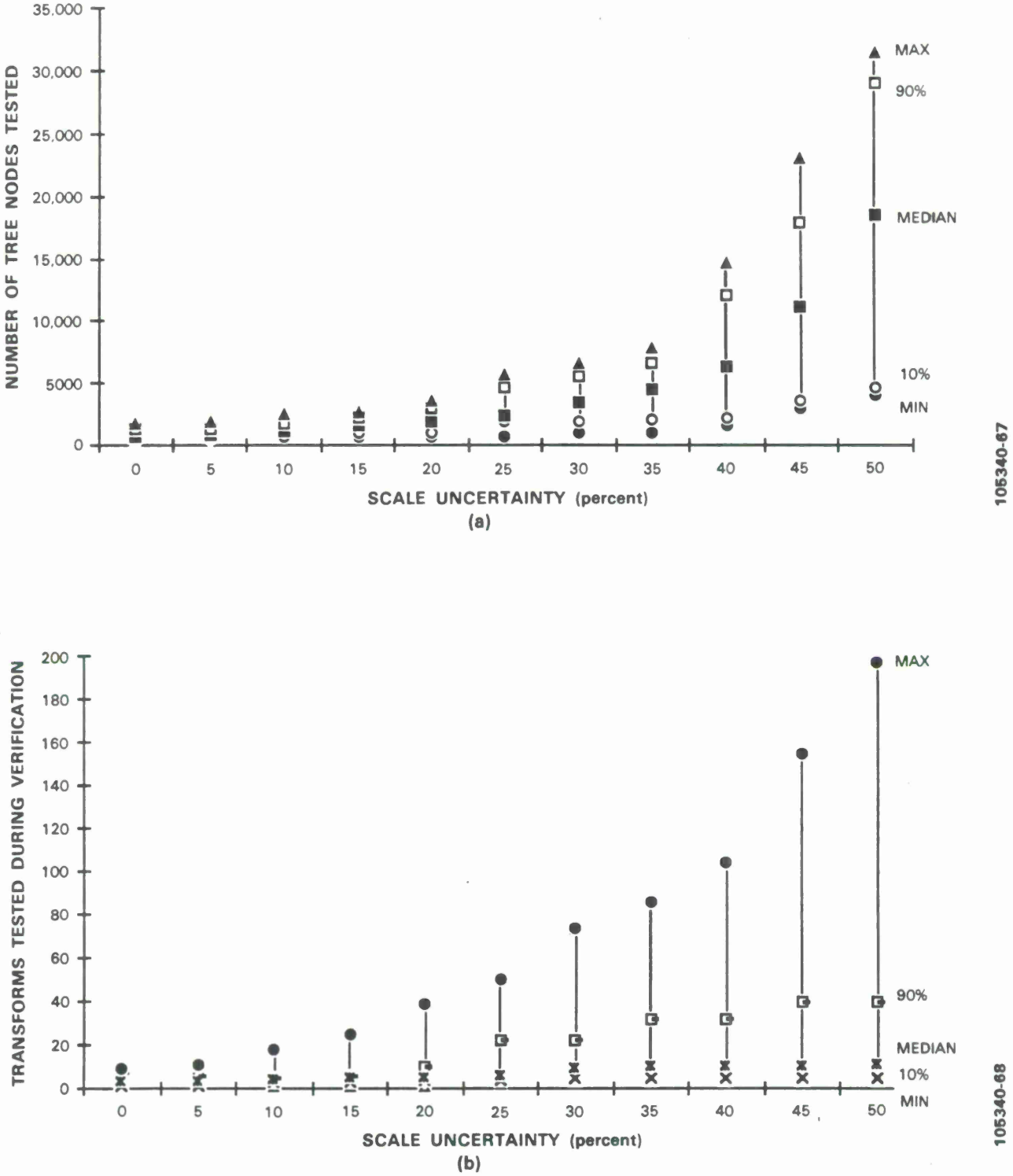
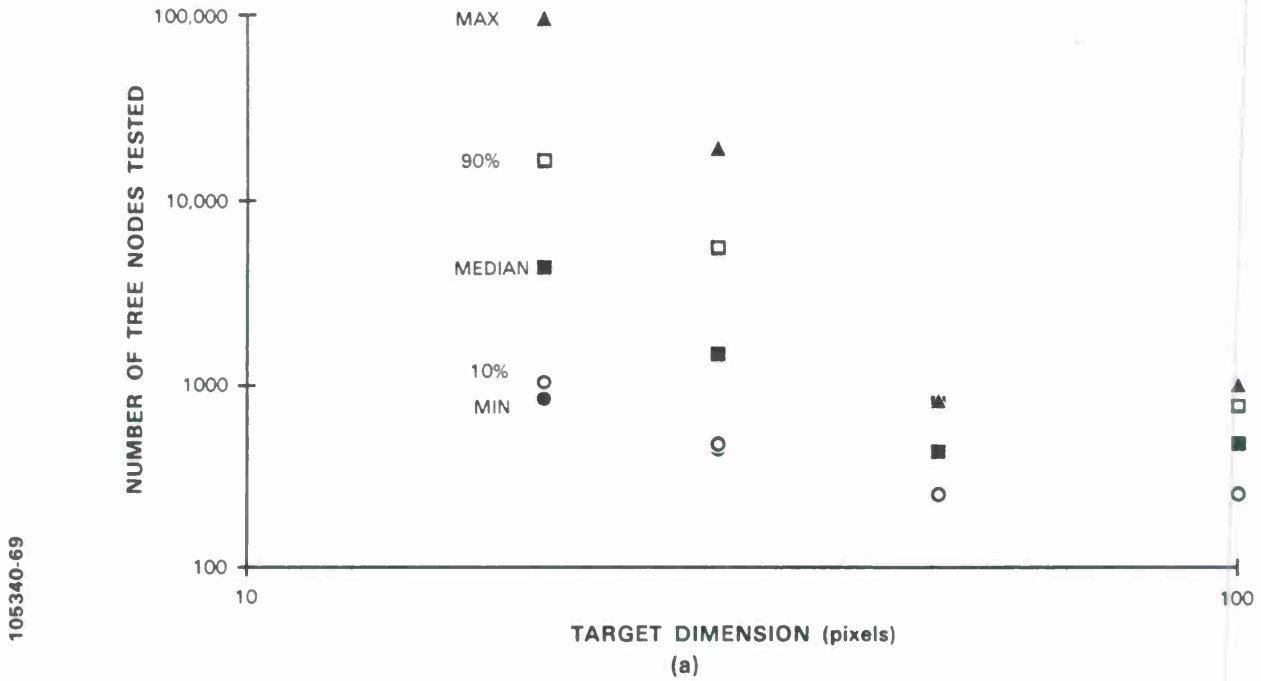


Figure 8-9. Effects of scale uncertainty.



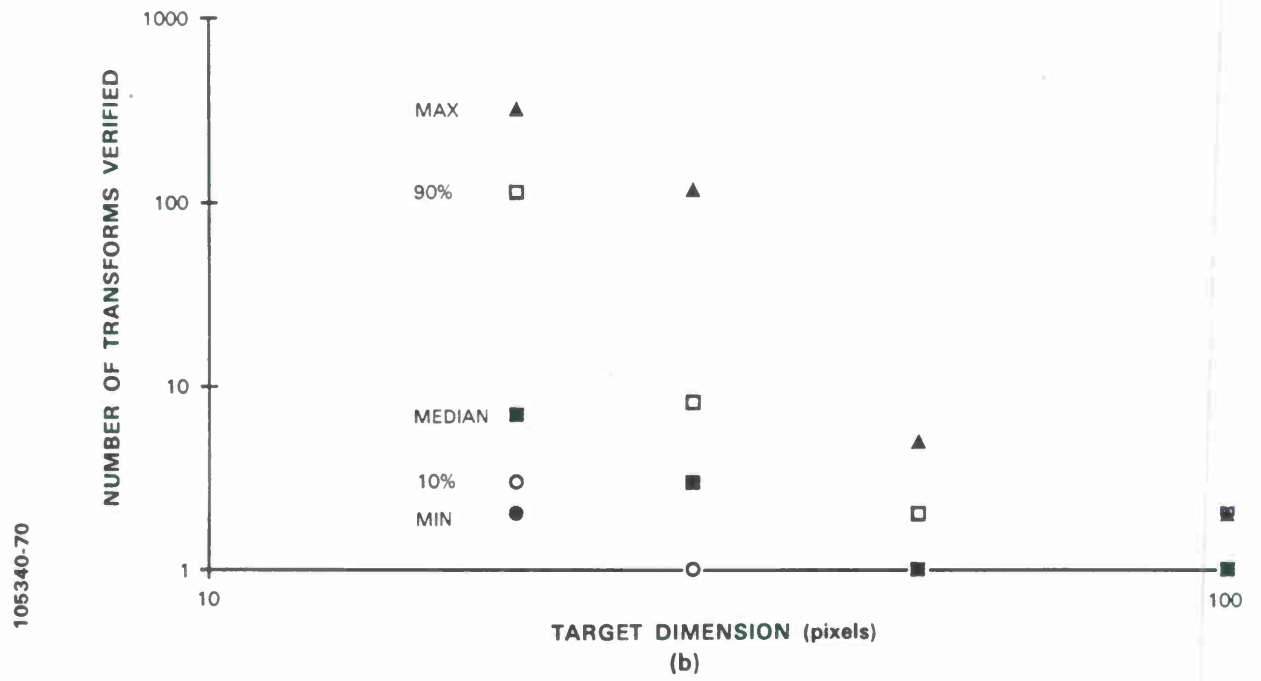


Figure 8-10. Effects of quantization.

number of interpretations consistent with the data. As a consequence, the tree search must test a large number of nodes and the verification must test large numbers of interpretations retained in the search. Note that in the experiments analyzed in these last graphs, the tree search was implemented separately from the verification so that the backtracking heuristic could not be exploited.

The third series of experiments evaluate the system performance when the tolerance on the number of unlabeled edges is increased. The graphs in Fig. 8-12 show again that the number of tree nodes tested and the number of verifications are substantially increased when the system must tolerate more unmatched edges. In the example, none of the silhouette edges had to be removed from the match. The numbers of null edges on the graph therefore indicate the numbers of extra null edges tolerated by the system. In our experiments, we have observed that when the system is given the correct number of null edges, the search is usually efficient. Problems arise mainly when the system tolerates many more null edges than necessary. Indeed, given a legitimate interpretation of the silhouette edges, other legitimate interpretations can be obtained by replacing any of the matched edges by the null edge. With excessive tolerances on the number of null edges and in the absence of backtracking heuristics, the presence of these multiple solutions

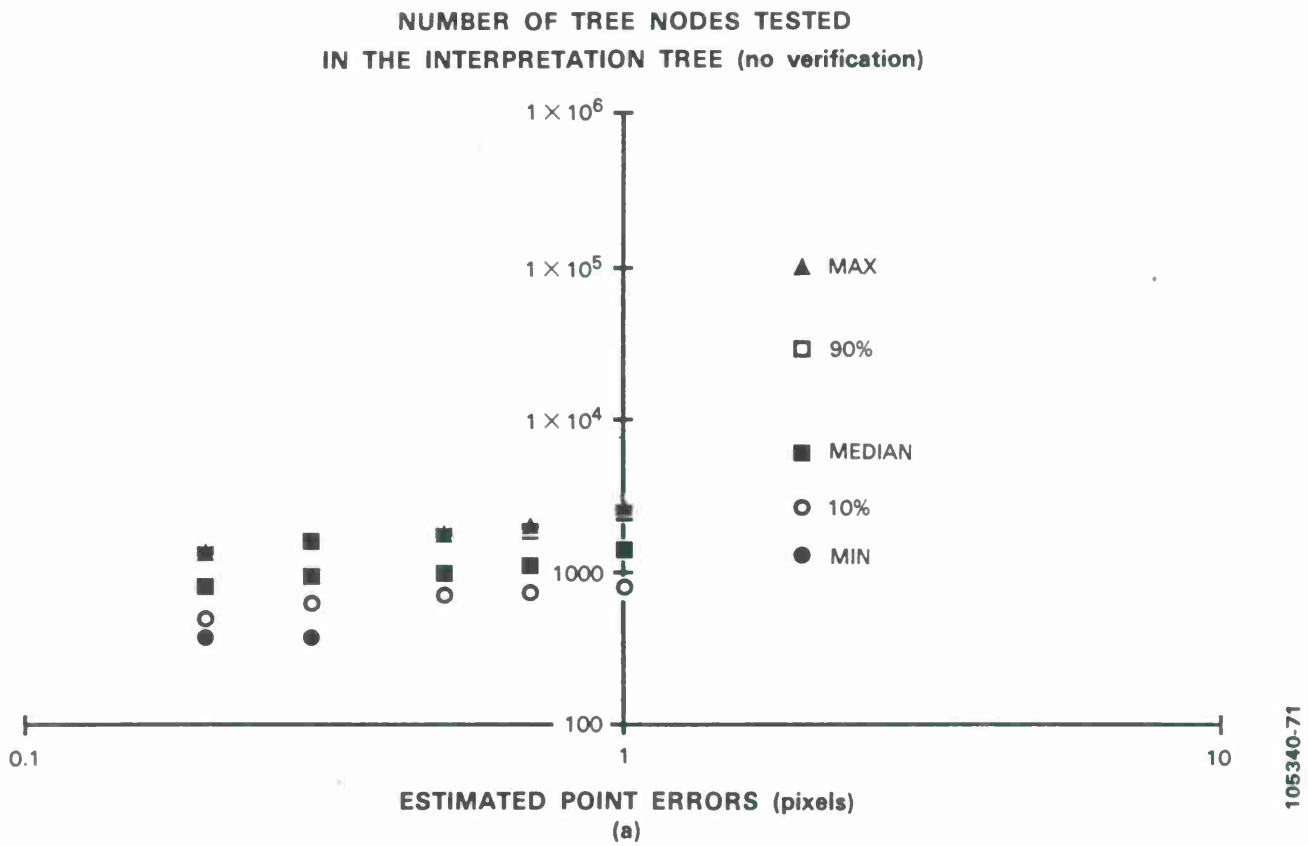


Figure 8-11. Effects of noise estimates.

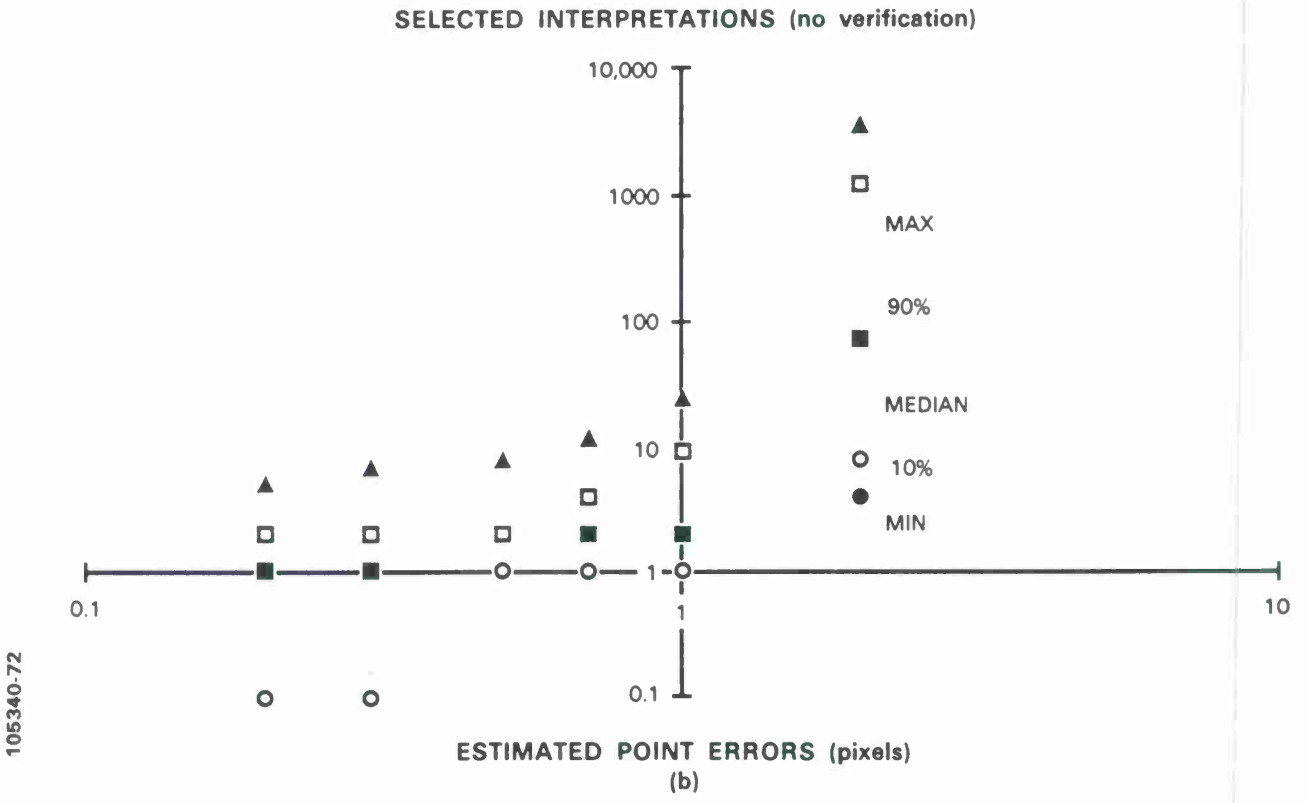


Figure 8-11 continued.

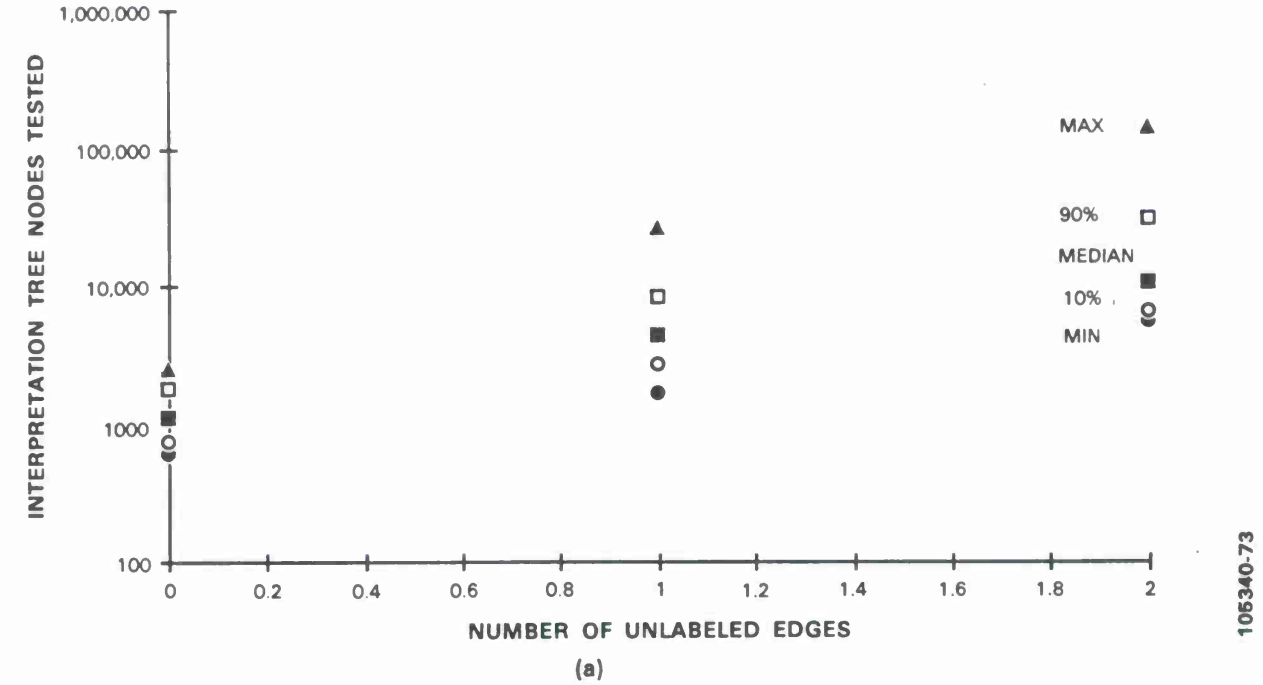
may hamper the system considerably.

8.3 UNRESOLVED ISSUES

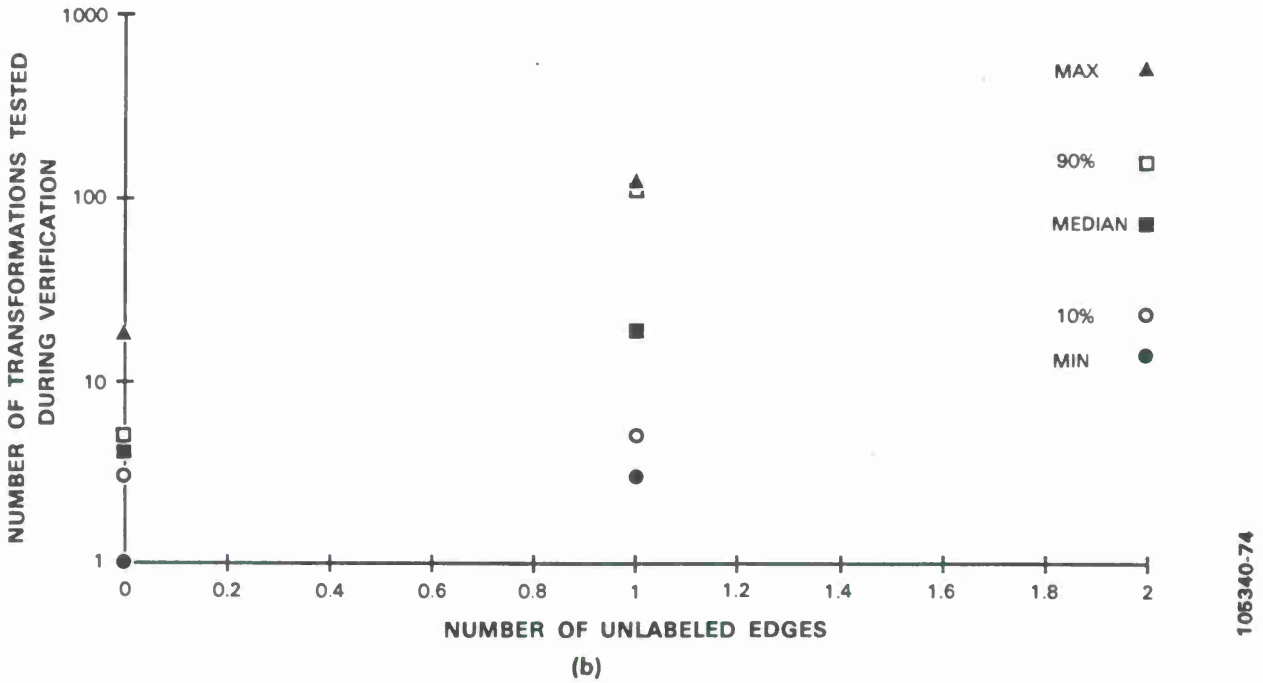
In this section, some of the major unresolved issues of the SILC system are discussed. Among these issues are excessive search times in cases of near-matches and spurious results in the presence of unstable object appearances. Another unresolved issue arises when the silhouette of interest is confused by large and unknown amounts of unrelated image details. This issue was addressed in Section 6.

8.3.1 Excessive Search With Near-Matches

In the current implementation, search times are kept to reasonable levels by the combination of heuristics presented in Section 6. Sometimes, however, the success of these heuristics is reduced by special configurations of the data and of the models, especially those where a data configuration almost fits the model for some viewpoint and for the given recognition parameters. This will occur, for example, when attempting to match a silhouette composed of multiple objects without allowing a sufficient number of unmatched edges. As we discuss below, the system may respond



(a)



(b)

Figure 8-12. Effects of unmatched edges.

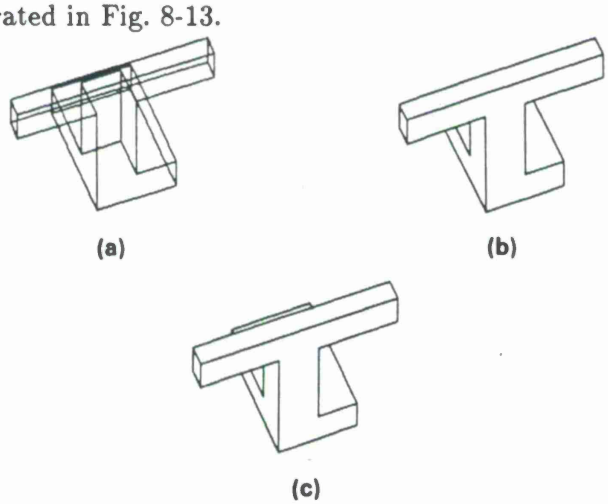
to a close mismatch by spending considerable effort in verifying that no combination of image edges will match the model.

In the presence of moderate to large numbers of smaller edges in the image, the expanded interpretation tree usually comprises branches with relatively large numbers of leaf nodes, as in Section 6.3.3. In order to avoid the excessive cost of performing a full verification on each leaf node of the same tree branch, an iterative optimization of the matching label was proposed in Section 6.3.3. This heuristic is extremely powerful when at least one of the leaf nodes corresponds to a correct interpretation, as it typically finds the best interpretation in 2 or 3 iterations. However, if none of the leaf nodes of a branch corresponds to a valid interpretation, the system will start a new iterative verification loop from each of these leaf nodes, in order to verify the absence of a correct match.

This situation will typically appear when the allowed number of unmatched edges is too low. Indeed, the correct solution (which correctly matches one image edge too few) combined with another edge match will typically fit the data sufficiently closely to be accepted in the tree search but not to be accepted after full verification. This situation can also arise when attempting a match with a model which closely resembles the original, or with a wrong orientation of a quasi-symmetric object.

8.3.2 Unstable Object Appearances

The issue discussed in this section is intrinsic to the matching of 2-D images with polyhedral 3-D models and concerns the difficulty of the problem for a non-generic viewpoint, that is a viewpoint where the topological nature of the silhouette may vary for small changes in viewpoint. A typical case is illustrated in Fig. 8-13.



106340-76

Figure 8-13. Viewpoint for which the silhouette is topologically unstable.

In this example (a), if the viewpoint is lowered by a small amount, the edge on top of the silhouette corresponds to the long edge on top of the model (b). However, if the viewpoint is raised by a small amount, the top of the silhouette is broken into three pieces corresponding to an edge

on the model top and one on its base (c). When presented with a raw silhouette corresponding to the viewpoint in (c), a vision preprocessing module may not detect the distinction between the three edges and merge them into one long edge. In that case, the tree search will select an interpretation of the image edges corresponding to (b). However, the viewpoint estimated from these matches may still be closer to (c) due to the other edges in the image. The synthetic silhouette has the top split into three edges and hence the match with the image edges is rejected.

This issue is not handled correctly by the SILC system, but it is really an issue intrinsic to the problem itself. It is likely that other systems performing a geometrical verification of the image edges with a synthetic silhouette, such as Lowe's SCERPO, would experience similar problems.

9. SUMMARY AND DIRECTIONS FOR FUTURE WORK

In this section, key results presented in the report are summarized. Directions for pursuing this work are then discussed.

9.1 SUMMARY

The SILC system presented in this report achieves new levels of performance in the recognition of silhouettes of 3-D objects with unknown orientation. We will first review system highlights, then discuss a number of key techniques exploited to attain this level of performance. These include the compilation of geometric appearance models, the interpretation tree search applied to the matching of 2-D images with 3-D models, and the development of precise models for propagating errors on edge measurements.

9.1.1 System Highlights

The SILC system decides whether an input silhouette matches a 3-D model in its database and, if successful, determines candidate positions and orientations of the 3-D object. It performs correctly in the absence of knowledge concerning relative positions and orientations of the object and the camera, in the presence of known levels of image noise and scale uncertainties, and in the presence of occluded object parts and extraneous objects.

The basic system always determines the correct interpretation of the given silhouette, i.e., its correct identity and location. In addition, it will also determine the interpretations of the silhouette for symmetric views of the same object, and even interpretations in terms of different object models, when the similarities of the models combine with the bounds on noise, scale and extraneous components to produce additional possibilities. When the models are highly symmetric and/or when the bounds on errors, scale and extraneous parts are weak, the number of valid interpretations may become large and the system response times degrades.

In the case of relatively simple objects and with low or moderate amounts of noise, the number of valid interpretations is usually limited to one or just a handful; these are generally discovered and verified in less than a second. A test of a silhouette with an incorrect model is usually rejected in less than a second. When the complexity of the models increases and/or when the estimated degradations of the input silhouettes become more severe, the number of valid interpretations usually increases and the system response degrades accordingly. In Section 6, we have described methods to limit the increase of system response times to acceptable levels for moderately complex objects, with moderate amounts of degradations; the acceptance or rejection of an interpretation is then performed in a few seconds at most. However, the improvement in response time is obtained by avoiding the exhaustive search of the interpretation tree. As a consequence, there is no longer a guarantee that the system will find the correct identification and location of the 3-D object, especially in the presence of multiple valid solutions. However, in our experiments, the

SILC system has always found the correct interpretation of the input silhouette in terms of the given models, when the degradations in the input don't exceed their estimated maximum levels.

9.1.2 Compilation of a Geometric Appearance Model

The object models are input to SILC as CAD-type descriptions of their geometry. However, this 3-D geometry is not directly related to the geometry of 2-D silhouette so it cannot be used as such during the interpretation tree search. The object geometry is first compiled into a set of constraint tables retaining the geometric appearance of each pair of edges in silhouette images of the corresponding object. The compilation consists of synthesizing silhouettes for a large set of viewing directions and of reducing the geometry of these silhouettes to a set of viewpoint-independent constraint tables. This approach can be compared to that of Weiss et al. [29] where each object is compiled into a set of models for its appearance in the image. However, major differences between the two approaches can be found. The Appearance Models of Weiss are symbolic whereas ours are geometric. In Weiss's approach, several models are defined, namely one for each "Characteristic View," whereas a single model represents the object for all viewpoints in our approach. Finally, the compilation of our models is completely automatic.

9.1.3 Interpretation Tree Search for 3D Vision from 2D Images

The interpretation tree search based on pairwise geometric constraints was demonstrated by Grimson and Lozano-Perez, first in the context of matching 2-D models with 2-D images (3 degrees of freedom), then for matching 3-D models with 3-D data (6 degrees of freedom but 3-D data). The application to matching 3-D models with 2-D data presented in this report goes one step further since the transformation between model and observation has five degrees of freedom and since the constraints provided by the 2-D data are not as powerful as those provided by 3-D data. The implementation of the constraints is performed by comparing relative positions of pairs of edges in the image with thresholds derived from the models. The image measurements are independent of the three degrees of freedom corresponding to image plane transformations. The remaining two degrees of freedom, corresponding to the viewpoint, are addressed by compiling model thresholds valid over the whole range of applicable viewing directions, in general over 4 π steradians, as discussed in Section 4.

Since the model thresholds must be valid over all viewpoints, the corresponding constraints are inevitably weaker than those applying to a single viewpoint. As a consequence, the pruning power of the constraints is weaker than for the 2-D/2-D identification, and appropriate measures must be taken to avoid an uncontrolled expansion of the interpretation tree (Section 6).

9.1.4 Precise Evaluation of Edge Errors

In order to reduce the effects of weaker constraints due to their validity over the complete range of viewing directions, we implemented a careful analysis of the propagation of errors from the estimation of edge element positions to the test of constraints on the relative positions of pairs of

edge elements. This analysis and the implementation of its results in the SILC system guarantee that the maximum constraint power is conserved in all circumstances. In contrast, measurement errors were considered in the system by Grimson and Lozano-Perez by estimating the bounds in the noiseless case and by simply allowing a fixed tolerance, uniform on all distances on one side and on all angles on the other side. With this scheme, a large tolerance must be used to ensure that correct matches will be accepted for the most extreme deviations of image measurements, but this large tolerance then decreases the constraint power for those measurements which are less affected by noise and biases.

9.2 DIRECTIONS FOR FUTURE WORK

The work presented in this report can and will be extended in a number of directions. In a first subsection, we consider direct applications of SILC as it is presented in this report; a second subsection addresses extensions of the capabilities and of the performance of the current system.

9.2.1 Direct Applications

The current version of SILC can be applied to any problem involving the recognition of 3-D objects given only silhouette images taken from unknown viewpoints. Two applications will be considered in particular, namely to the recognition of space objects from Range-Doppler radar images, and to the recognition of targets in laser radar images.

Range Doppler Image Understanding

The identification of man-made space objects from Range-Doppler images is a particularly appropriate application for this system. First, the objects are imaged at a large distance, thereby guaranteeing orthographic projection. Second, there is often no a-priori information about the orientation of these objects. Finally, satellites and other space objects usually exhibit large and well-defined geometrical structures such as solar panels. The geometry of these structures in silhouettes is readily recognized by SILC. The difficulty in this application will probably come from the relatively poor quality of the imagery. Substantial smoothing of the silhouette contours may be necessary to prevent excessive splitting of the longer edges that best characterize the object shape.

Laser Radar Image Understanding

The identification of targets in laser radar images will reveal the degree to which SILC can tolerate approximate models, severely subsampled images, and the presence of clutter. In this application, too, the targets are usually located at a large distance from the sensor, justifying the orthographic projection approximation. In a large fraction of the imagery, a-priori restrictions on the position and orientation of the target are available and could be exploited by the system;

this is especially true for ground-based imagery where only small deviations of object orientations from the vertical can be expected. These a-priori constraints can be exploited in SILC by considering only the corresponding set of viewing directions during the compilation of pairwise edge constraints. Note that in the current implementation, the a-priori constraints must be known during the compilation of the models. Although models could be recompiled at run-time, given restrictions on the viewing direction, the compilation time (several minutes in the current system) makes this choice undesirable.

9.2.2 Further Developments

During the development of SILC, we have concentrated on basic features to demonstrate recognition of rigid objects based on the interpretation tree search with simple pairwise edge constraints. However, the system concept can be extended in a number of ways discussed below. These include using additional features extracted from the image and their model correspondents, investigating the application to recognizing of articulated objects, and using heuristics to speed the tree search.

Extension of Features

In the current implementation of SILC, only silhouette edges extracted from the image can be exploited in the identification and the estimation of the orientation. Except for increasing the system complexity, there is little difficulty in extending the range of features which can be used with the same system concept. We first consider extensions to current implementation of edge features, and then we briefly consider extending the system to incorporate other types of features.

Allowing 180 Degrees Ambiguity in Normal Orientation SILC currently requires that the direction of the silhouette edges be provided. Although the inside/outside direction is easily identified for backlit objects and in range imagery, determining the outward normal may not be easy for all grey scale images. This restriction can be avoided by trying in turn both outward normal directions for each image edge. This is implemented for example by expanding each tree node into $2M + 1$ subnodes, corresponding to the M model edges taken each for both normal orientations of the image edge and the null edge, instead of $M + 1$ subnodes in the current system.

Exploiting Interior Edges The current system exploits edges only on the silhouette in the image plane. It is possible, however, to include image edges interior to the silhouette with a few modifications of the system. First, the current constraint tables are determined by compiling synthetic silhouettes of the object models. Different constraint tables must be built to incorporate all visible edges in the object image.

This simple approach has a disadvantage. Indeed, the visibility of each edge is greatly increased by considering it both on the silhouette and as an interior edge. As a result, the range of configurations of each pair of edges is vastly increased and therefore, the constraint power is

reduced. In order to offset this disadvantage while still retaining the larger support of all image edges, it is possible to define two different appearances for each model edge, namely one on the silhouette and one as an interior edge. When possible, each image edge is classified either as a silhouette edge or as an interior edge, and matched only to the corresponding model edge appearance. Image edges for which the classification is uncertain can be matched to either a silhouette appearance or an interior edge appearance.

Considering interior edges is likely to raise a number of implementation details. First, although a silhouette edge can be oriented with the outward normal, there is a 180 degrees ambiguity in the orientation of interior edges. A different issue is that for a non-convex object, the same model edge can appear partially on the silhouette and partially on the interior for the same viewpoint. If the two parts are correctly separated and identified, the match will succeed, but an issue may arise if the image edge is extracted in one piece.

Other Features The general framework implemented in the SILC system can be extended to cover any type of feature that can be readily extracted from the image, when the image appearance can be synthesized from a model of the 3-D object. For example, it would be possible to design a system based only on silhouette corners, or a system based on the combination of edges and corners. Constraint tables for the models and for the images would be computed in the same way as in the current SILC system, but the matching would consider only correspondences between similar features.

The general strategy exploited in SILC, namely the interpretation of image features with a tree pruned by testing geometric constraints, can be applied to a large variety of signal understanding problems. One might consider applying similar strategies to target identification in radar images based on the geometry of reflector returns.

Articulated Objects

It is likely that the scope of the SILC system can be extended beyond the set of rigid objects modeled by polyhedra, for example by considering articulated objects such as a pair of scissors. A possible implementation of a SILC type system for articulated objects considers model constraint tables compiled not only for all object orientations but also for all angles of the articulations, more generally for all values of the internal parameters. The search stage is then performed as usual. In the verification stage, both object orientation and values of the internal degrees of freedom must be estimated from the correspondences predicted in the search phase. This last issue has been addressed successfully by Goldberg and Lowe [9].

Search Heuristics

One of the disadvantages of the SILC system is the potentially exponential complexity of the tree search phase. Although the algorithm is usually completed in matters of seconds for segmented objects, the search may be extremely slow for poorly segmented scenes. In the SCERPO

system, Lowe avoids the exponential complexity of matching in the absence of segmentation by exploiting heuristics to quickly relate a small number of image edges with the appropriate model edges. However, the faster processing times are compromised with respect to the generality of the system; the fast pairing is possible only for model edges in special configurations, such as parallel or colinear edges. In addition, the SCERPO system attempts to find one solution to a recognition problem, while the SILC system tries to discover all valid interpretations.

A strategy combining the fast processing times of SCERPO and the exhaustive search of SILC in a single system could be obtained by applying the SILC strategy after sorting individual model edges and image edges according to the same heuristics used in SCERPO. This merge of the two strategies would raise non-trivial issues since the heuristics for ordering model edges would depend on the matches already hypothesized. As a result, the next silhouette edge to be tried and the ordering of the model edges would depend on the current position in the tree. However, if these issues can be solved, the resulting system could be tuned according to the user needs to balance the advantages of the SCERPO strategy and of the SILC strategy.

APPENDIX A
SILHOUETTE PARSING

In the SILC system, a silhouette is matched to a 3-D model by comparing descriptions of their shapes in terms of straight edges. This appendix reports on a number of methods developed for analyzing a silhouette and for extracting a description of its shape in terms of a set of straight edges.

The shape of a silhouette can be represented by a collection of straight edges in a number of different ways, and the particular representation must be chosen according to the role that this representation is to fulfill. In this work, the straight silhouette edges are exploited primarily in the search of the interpretation tree of the silhouette shape in terms of model shapes. We have considered representing silhouettes by polygonal approximations of the contour such as those reported in [7,17,20,27], and by descriptions based on the labeling of contour points in terms of straight edges, curves, and corners. By employing both theoretical and statistical analysis of the tree-search algorithm performance on a moderately large data set, we have determined the qualities and disadvantages of the different parsing schemes and selected two particular choices for the SILC system.

A.1 APPROXIMATING A CONTOUR BY STRAIGHT EDGES

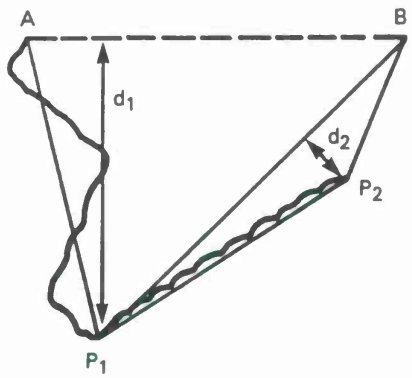
In this section, we review a number of methods for analyzing the shape of a silhouette and for describing it with a set of straight edges. In the first method, referred to as polygonal approximation, the silhouette is modeled by a sequence of contiguous edges; with this method a closed silhouette is modeled by a closed polygon. The exact shape of this polygon is determined by optimizing its fit with the original contour, subject to a number of constraints. Although several criteria have been developed for determining the fit, we will investigate only one based on maximum deviation, which was first reported by Ramer [20] in this context.

In the second method, the silhouette shape is analyzed by a set of local shape estimators to determine whether each silhouette point is on a straight edge, a curve, or a corner. The contour model is then built from the results of this analysis. In the absence of noise, local operators can easily determine which parts of a silhouette correspond to straight lines, corners, and curves. When the data is corrupted by noise and other artifacts, however, the performance of local estimates becomes poor. Better estimators for noisy data either work locally on a smoothed version of the silhouette or perform their estimates on some neighborhood of the silhouette. The increase robustness of these operators with respect to noise is obtained by compromising the accuracy of the contour characteristics. We will first discuss local estimation of slopes and curvatures, then estimation based on neighborhoods and smoothing of the silhouette.

A.1.1 Ramer Polygonal Decomposition

The Ramer polygonal approximation represents a given curve by a polyline in such a way that the distance between any point of the curve and the approximating polyline is less than a predetermined tolerance. In this method, the vertices of the polyline, which are called breakpoints, are chosen on the given curve. The appeal of the method lies in its relatively fast implementation, its recursive structure, and its intuitively appealing results. However, the method does not claim any optimality in terms of the maximum error or the number of edges required. We will first discuss the case of open curves, then turn to the case of closed curves.

For any given curve, different values of the tolerance will determine a series of Ramer decompositions with an increasing accuracy as the tolerance is reduced. For a very large value of the tolerance, an open curve AB is approximated by the segment AB joining its two endpoints, as illustrated in Fig. A-1.



106340-76

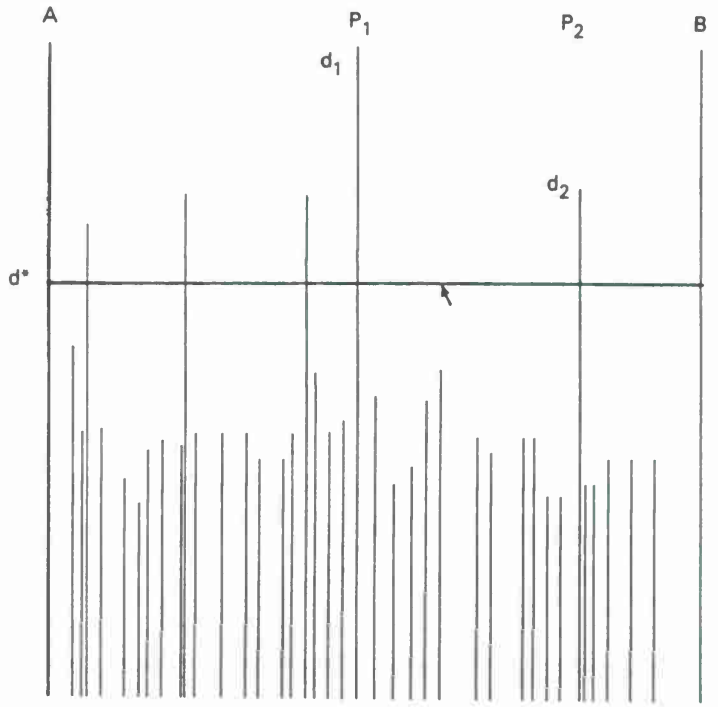
Figure A-1. Ramer decompositions of an open curve.

When the tolerance is decreased, the same decomposition applies until the tolerance becomes smaller than d_1, the largest deviation of the curve from the segment corresponding to the point P_1 on the curve. When the tolerance is decreased below d_1, the curve is approximated by the polyline AP_1B; this decomposition is valid until the tolerance decreases below d_2, the largest deviation between the second approximation and the curve, corresponding to P_2. The curve is modeled by AP_1P_2B for tolerances just below d_2. The polygonal decomposition for a given tolerance is constructed by proceeding with the recursive decomposition of each segment of the approximation by the point of the curve most distant from the segment, until the distance of the candidate breakpoint is smaller than the given tolerance.

The Ramer Spectrum

It is interesting to consider the set of all Ramer decompositions of a given curve for values of the tolerance from 0 to ∞. First, for a discrete contour, the set is finite and its size is one less than the number of points on the curve, in the absence of alignments. Each point of the curve becomes a breakpoint for some value of the tolerance and remains a breakpoint for all tolerances

smaller than this value. As a consequence, the set of all decompositions can be represented by a graph such as illustrated in Fig. A-2.

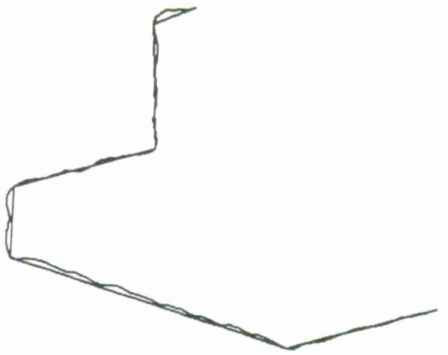


105340-77

Figure A-2. Ramer spectrum of the curve in Fig. A-1.

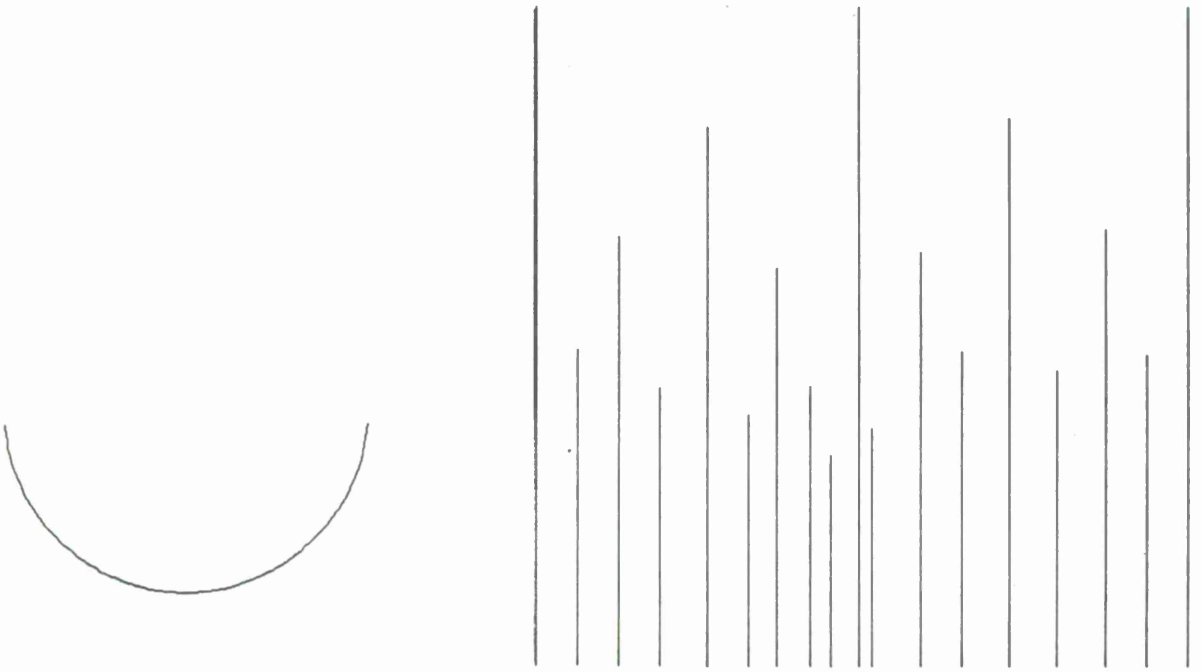
This figure is a plot of the tolerances for which each point of the curve becomes a breakpoint. Each vertical bar corresponds to a particular point on the curve; the tolerance corresponding to the top of the bar is the maximum tolerance for which the point is a breakpoint in the decomposition of the curve. A horizontal slice through the spectrum is closely related to the decomposition of the curve for a tolerance given by the height of the slice. Indeed, the intersections of the horizontal line with vertical bars of the spectrum correspond to the breakpoints for that particular tolerance. For the tolerance d^* shown on Fig. A-2, there are seven breakpoints, with the resulting decomposition as shown in Fig. A-3.

We will refer to the plot in Fig. A-2 as the Ramer spectrum of the curve and to the value of the critical tolerance for each point as the Ramer spectrum value at that point. The Ramer spectrum is useful for describing the shape of an object at various degrees of accuracy, and for guiding the choice of an appropriate tolerance for the decomposition. The Ramer spectrum will also be useful in the discussion of Ramer decompositions for closed curves. It is easy to see that tall bars in the spectrum correspond to "important points" of the curve which usually correspond to the major corners of the curve. The heights of the impulses are correlated with the importance of the associated corners in the characterization of the silhouette shape. Note that the Ramer spectrum of a circle displays the effects of a regular recursive subdivision, as shown in Fig. A-4.



105340-78

Figure A-3. Decomposition of the curve in Fig. A-1 for the tolerance d^*.



105340-79

Figure A-4. Ramer spectrum for a half circle.

Ramer Decomposition for Closed Curves

A closed curve can be decomposed with the technique developed for open curves when it is considered as an open curve with coinciding endpoints. However, this procedure requires the choice of a particular point of the curve to start the decomposition. A possible choice of starting point is an endpoint of the largest diameter of the curve. This choice is appealing since the first non-trivial decomposition corresponds to the largest diameter of the curve and since this choice is independent of position and orientation. We have observed that the decomposition starting from the largest diameter was satisfactory for polygonal convex objects, but that it did not always include smaller details of concave objects or properly described curves. We have therefore

developed an alternative decomposition which we refer to as the "isotropic Ramer decomposition."

The Ramer decomposition of a closed curve depends on two parameters, namely the starting point of the decomposition and its tolerance. Conceptually, the set of all Ramer decompositions is then equivalent to a set of Ramer spectra: one for each choice of the starting point. Except for the starting point itself and for small variations, all these spectra are quite similar for a convex polygonal silhouette. However, curves are decomposed differently when the starting point is modified, and the value of the spectrum for at a particular point on the curve may vary significantly in some cases.

We have defined the "isotropic Ramer spectrum" of a closed curve as a summary of the simple Ramer spectra of the curve corresponding to all possible starting points. The value of the isotropic spectrum at a given silhouette point defined as is the maximum value of the simple spectrum for this point constructed for all starting points on the contour. We then define the isotropic Ramer decomposition of a silhouette for a given tolerance by its breakpoints . The breakpoints correspond graphically to the intersections of the isotropic spectrum with a horizontal line corresponding to the tolerance.

Comparison between Uniform and Simple Ramer Decompositions We have developed an interactive system for exploring decompositions of silhouettes based on the isotropic Ramer spectrum. Using this system, we have conducted extensive experiments on various shapes of silhouettes. Typical examples of the isotropic Ramer spectrum are shown in Fig. A-5 for a polygonal silhouette and for a semi-circle.

The structure of the isotropic Ramer spectrum of a polygonal silhouette is very similar to that of any of its simple spectra. The isotropic Ramer spectrum of a polygonal silhouette with rounded corners contains broad peaks at the corners, while the simple Ramer spectrum contains a single peak. Curves lead to bands of high spectral values in the uniform spectrum, as opposed to the recursive subdivisions of the simple spectrum. When comparing the uniform decomposition of a silhouette with a simple Ramer decomposition for the same tolerance, the former usually has more breakpoints and provides a closer approximation of curves. In the frequently encountered case of a polygon with rounded corners, the simple decomposition usually selects a breakpoint in the vicinity of each important corner, as shown in Fig. A-6.

The deviation between the approximation and the observed silhouette will usually be on the same order of magnitude as the tolerance. For the same tolerance, the uniform decomposition will decompose the corner into several small segments and will provide a much better approximation of the adjacent edges, as illustrated in Fig. A-7. Although the accuracy of the uniform decomposition is usually better than for the simple decomposition, the error bounds are difficult to determine in both cases.

The computation of an isotropic Ramer decomposition is more expensive than a simple spectrum by a factor approximately equal to the number of points on the curve. It is hence important to consider whether the increase in complexity is justified by improvements in the results, or if comparable results can be obtained by a different technique. As mentioned earlier, an isotropic

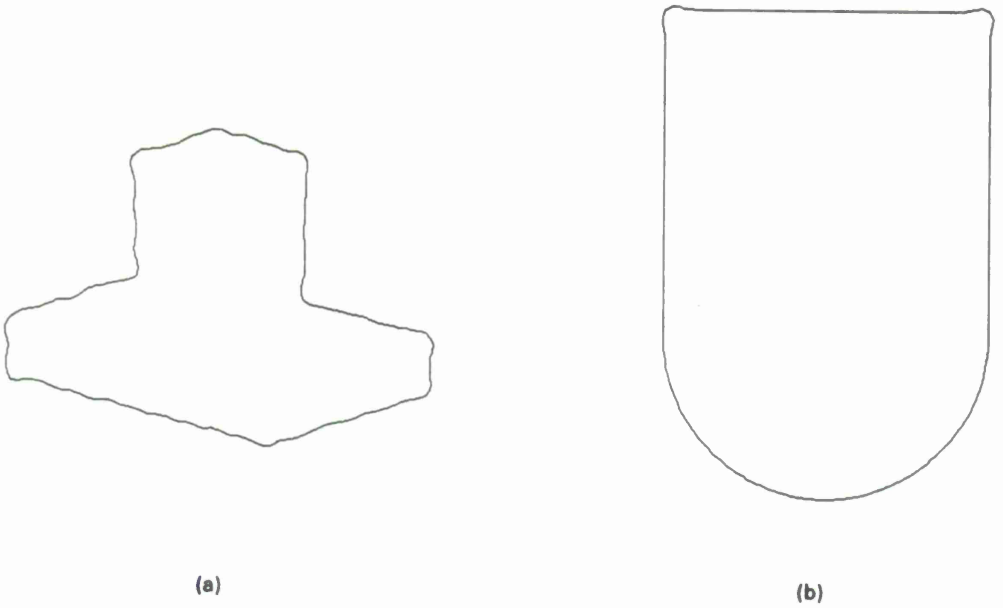


Figure A-5. Ramer spectrum comparison for (a) polygonal and (b) semi-circular silhouettes.

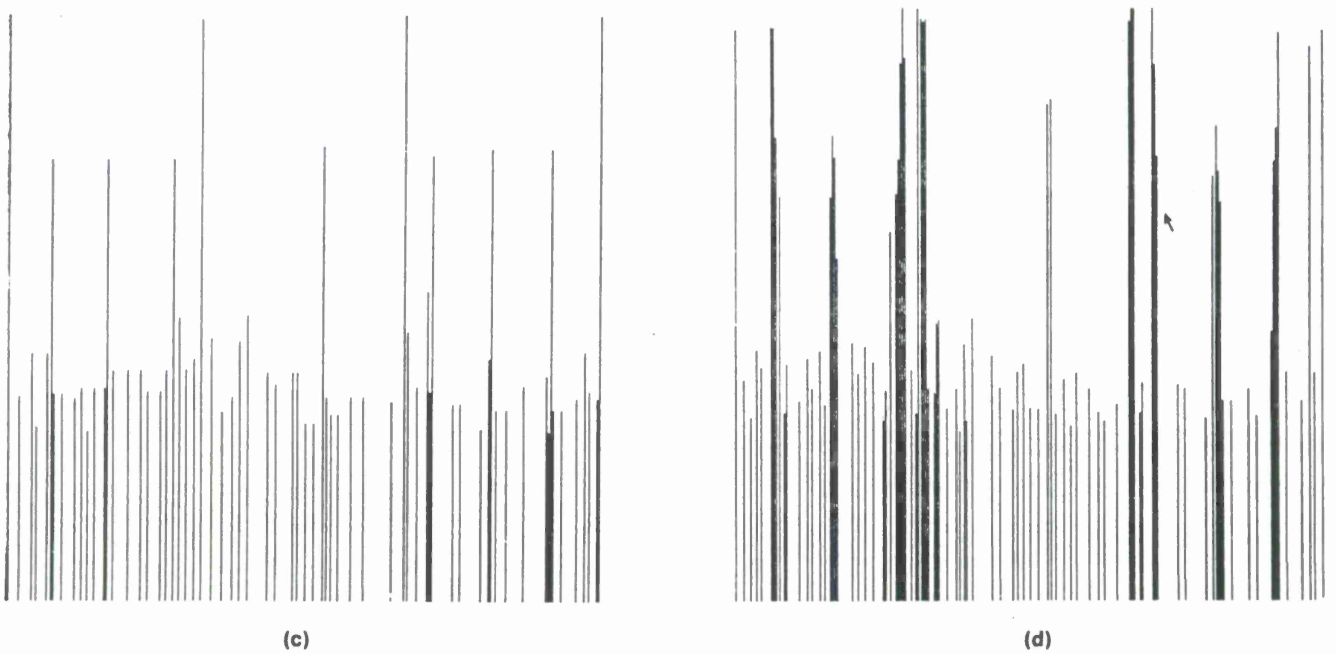


Figure A-5 continued. Simple (c) and isotropic (d) spectra for the polygon in (a).

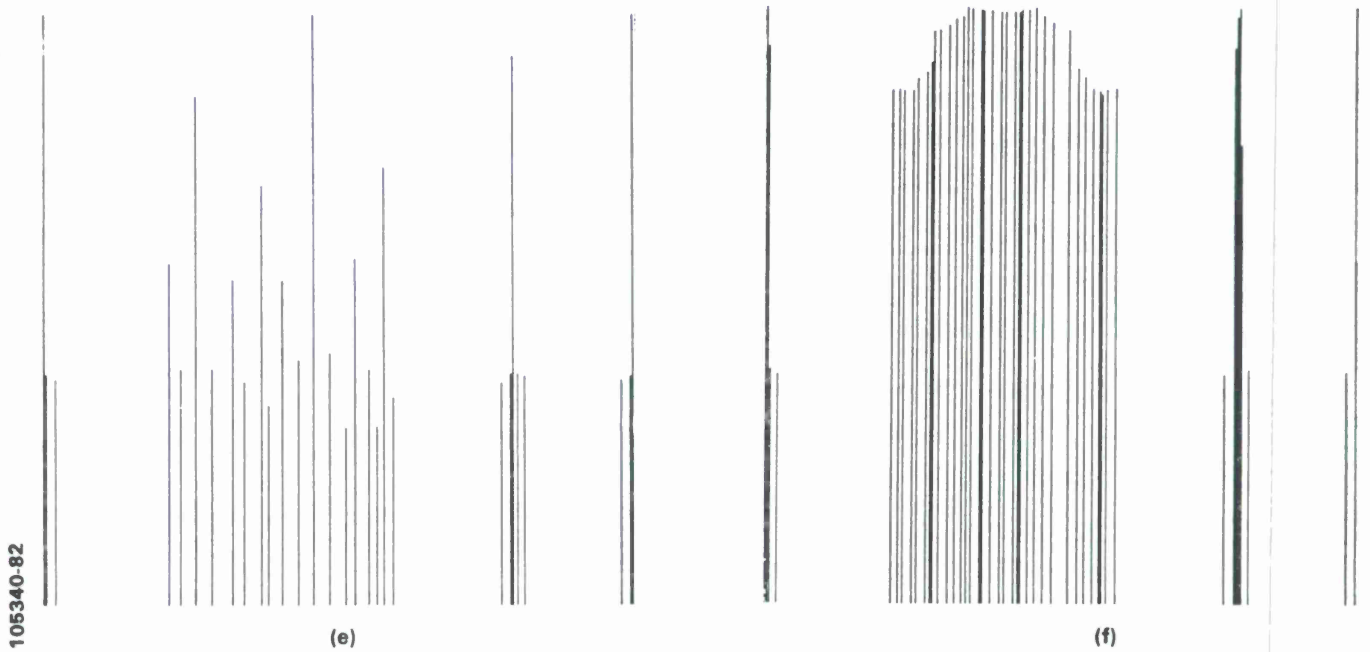


Figure A-5 continued. Simple (e) and isotropic (f) spectra for the polygon in (b).

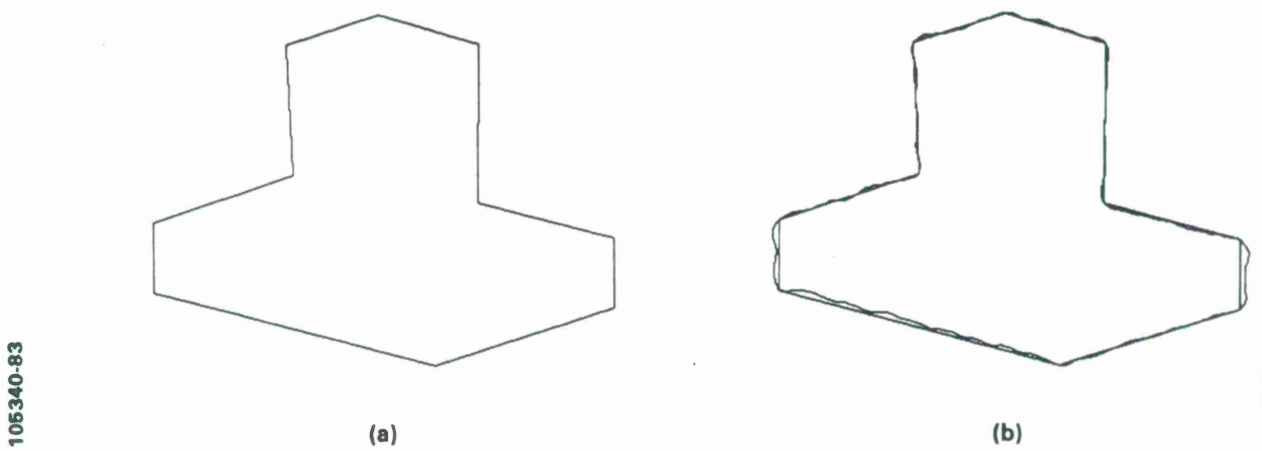


Figure A-6. Simple decomposition of a polygonal silhouette with rounded corners (a), also shown superimposed on the original (b).

Ramer decomposition is independent of the starting point and therefore independent of the orientation of the curve. In a majority of cases, however, simple Ramer decompositions for a tolerance smaller than the size of the object by an order of magnitude are relatively independent of the starting point, except for the starting point itself. Another advantage of the isotropic Ramer decomposition is its improved handling of corners and curves. Although the results show sub-

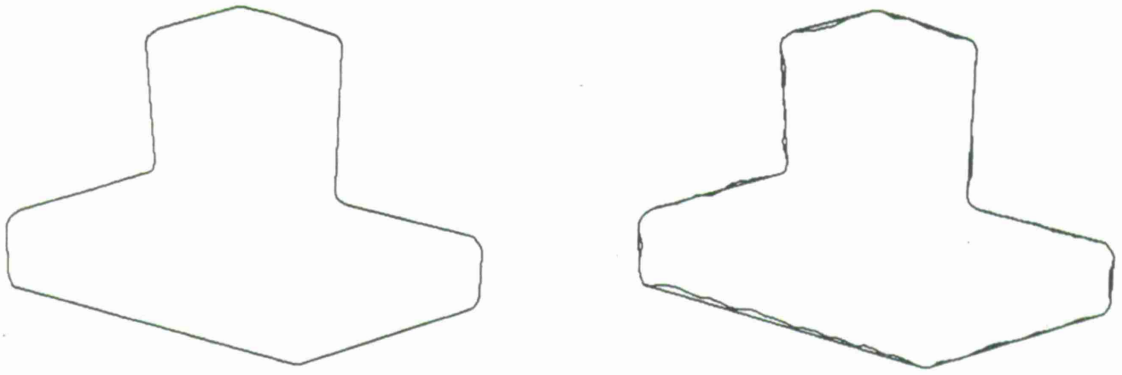


Figure A-7. Isotropic ramer decomposition of the silhouette in Fig. A-6.

stantially improved accuracy over the simple Ramer decompositions with the same tolerance, the advantage over a simple decomposition leading to the same number of breakpoints is inconclusive. Owing to the above observations, we have retained the simple Ramer decomposition in SILC.

In the SILC system, the description of silhouette edge chains provided by the edge extraction functions include a starting point at the highest point of the silhouette in the image. As this point is likely to become a breakpoint for relatively large values of the tolerance, this point is used as a starting point for the simple Ramer decomposition; this decomposition generally produces a good description of the silhouette.

A.1.2 Decomposition Based on Estimates of Slopes and Curvatures

In this section, we discuss a decomposition of silhouettes into straight edges based on local estimates of the silhouette shape. This decomposition relies heavily on methods for estimating the slope and curvature at a point of a silhouette. The first several methods presented provide local estimates of the slope and curvature assuming that the silhouette is free of noise. Then we will present two well-known estimation methods for noisy silhouettes. The first method derives its estimates at a given point from an extended neighborhood of that point, whereas the second uses local estimates on a smoothed version of the silhouette. These methods produce identical results in some cases.

Pointwise Estimates of Slope and Curvature

The slope of a continuous curve can be defined as the angle of the tangent with the horizontal axis; for a curve defined by the vector parametric equation $\vec{x} = \vec{x}(t)$, the slope $\phi(t)$ is

$$\phi(t) = arg(dx/dt + jdy/dt) \tag{A.1}$$

where $arg()$ denotes the polar angle of the complex number argument. For a discrete curve $\vec{x} = \vec{x_i}$, the slope is given by

$$\phi_i = arg(x_{i+1} - x_i, +j(y_{i+1} - y_i)) \tag{A.2}$$

Note that the slope index has an offset of 1/2 with respect to the curve, i.e., ϕ_i denotes the slope between points indexed i and $i+1$.

The curvature k of a continuous curve can be defined as the inverse of the radius of curvature of the osculating circle; it is also equal to the derivative of the slope with respect to the arclength.

$$k(t) = \frac{d\phi}{ds} = \frac{d\phi/dt}{\sqrt{dx/dt^2 + dy/dt^2}} \tag{A.3}$$

For a discrete curve, the true curvature is either zero or infinite, but a measure equivalent to a curvature can be defined by adapting the definition for smooth curves. A first possible definition is the inverse radius of curvature of a circle through the point and its two neighbors.

$$4k_i^{-2} = [x_i - x_{i-1} + \lambda(y_{i-1} - y_i)]^2 + [y_i - y_{i-1} - \lambda(x_{i-1} - x_i)]^2 \tag{A.4}$$

with

$$\lambda = \frac{(x_{i+1} - x_{i-1})(x_i - x_{i+1}) + (y_{i+1} - y_{i-1})(y_i - y_{i+1})}{(y_{i-1} - y_i)(x_i - x_{i+1}) - (x_{i-1} - x_i)(y_i - y_{i+1})}$$

Another definition of the curvature can be designed as the difference in the slope of contiguous segments divided by the length of the segments. As noted earlier, the slope is equivalent to the argument of the complex number $(x_{i+1} - x_i) + j(y_{i+1} - y_i)$; therefore, the change of slope is equivalent to the argument of the conjugate product of the complex numbers corresponding to two successive segments.

$$k_i = \frac{arg\left[\begin{array}{l}(x_i - x_{i-1})(x_{i+1} - x_i) + (y_i - y_{i-1})(y_{i+1} - y_i) \\ +j((x_i - x_{i-1})(y_{i+1} - y_i) - (x_{i+1} - x_i)(y_i - y_{i-1}))\end{array}\right]}{1/2\left[\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} + \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}\right]} \tag{A.5}$$

The above formula is preferred over A.4 when simultaneously estimating both slopes and curvatures. The results of these two methods differ by less than 10^{-3} when the points span less than a 15° slope differential on the curve.

Local Estimates of Slope and Curvature on a Noisy Silhouette

The slope and curvature estimators presented in the previous subsection produce accurate results for noiseless data; however, their estimates are based on two or three data points only and are therefore extremely sensitive to noise. A general strategy for reducing the variance of a noisy estimate is to consider averages over several data points. This strategy can be applied to our problem in three different ways. The first consists of smoothing the noisy estimates by a low-pass filter. The second consists of applying our estimators to a smoothed version of the silhouette. The third consists of developing estimators based on larger sets of points. The first approach is a trivial application of signal processing to the output of the estimators developed in the previous section. Estimators based on larger numbers of points are discussed in the present section, whereas smoothing of the silhouette and the slope and curvature estimates on the smoothed silhouette will be discussed in the following section.

Estimation of Slope In order to estimate the slope at a point i_0 of a noisy silhouette, we consider a set of $2N+1$ points centered at i_0, and use for the estimate, the slope of the line best fitting the $2N+1$ points. The best fitting line is the line $ax+by+c=0$ which minimizes the errors in fitting the datapoints, i.e.,

$$e_i = ax_i + by_i + c \tag{A.6}$$

When the objective is to minimize a weighted sum of squares of the errors, $E = \sum_{i=i_0-N}^{i+N} w_i e_i^2$, the parameters of the best fitting line are given by

$$\begin{aligned} a &= \sin\theta \\ b &= \cos\theta \\ c &= -ax_g - by_g \qquad \text{(A.7)} \\ \theta &= 1/2 arg(S_{xx} - S_{yy} + j(-2S_{xy})) \\ E &= a^2 S_x x + 2ab S_x y + b^2 S_y y \qquad \text{(A.8)} \end{aligned}$$

where (x_g, y_g) is the center of mass and $S_x x$, $S_x y$, $S_y y$ are the centered moments of the weighted data points. The angle θ can be used as an estimate of the slope at the point i.

Estimation of Curvature In order to estimate the curvature at a point i_0 of a noisy silhouette, we consider the set of $2N+1$ points centered at i_0, and use the curvature of the circle which best fits the data points. The best fitting circle is the circle $(x-x_0)^2 + (y-y_0)^2 = R^2$ which minimizes the deviations in fitting the data points, i.e.,

$$e_i^2 = (x_i - x_0)^2 + (y_i - y_0)^2 - R^2 \tag{A.9}$$

When we minimize the sum of squares of the deviations, $E = \sum_{i=i_0-N}^{i+N} e_i^2$, the parameters of the best fitting circle are obtained by solving

$$\left[\begin{array}{lcl} x_0 \sum(x_i - x_g)^2 + y_0 \sum(x_i - x_g)(y_i - y_g) & = & 1/2 \sum(x_i^2 + y_i^2)(x_i - x_g) \\ x_0 \sum(x_i - x_g)(y_i - y_g) + y_0 \sum(y_i - y_g)^2 & = & 1/2 \sum(x_i^2 + y_i^2)(y_i - y_g) \\ 1/(2N+1)\left[\sum(x_i - x_0)^2 + \sum(y_i - y_0)^2\right] & = & r^2 \end{array}\right. \quad \text{(A.10)}$$

Note that the above solution [1] does not directly minimize a sum of squares of distances between points and the circle, but this deviation is necessary to develop a closed-form solution.

Examples Figure A-8 shows examples of slopes and curvatures obtained with the algorithms described above, using three different values N, the half window size. It appears from these diagrams that, as N increases, the variability of the estimates decreases, but that the estimates are increasingly biased by smoothing. Indeed, when estimating local parameters of a signal corrupted by random noise, there is always a trade-off between accuracy and locality of the estimates on one side, and the signal to noise ratio on the other side. The compromise can often be improved by using tapered weights on the set of points used to estimate the parameters at each point.

In the next section, we discuss an alternative method for estimating the slope and curvature, namely one that first smooths the silhouette by averaging over a set of data points, then performs the estimates on the smoothed curve. This method provides a good flexibility in the design of the weighting assigned to different data points. In the case of slope estimation, the results are comparable to the ones obtained with the method presented earlier.

Estimates of Slope and Curvature on a Smoothed Silhouette

In this section, we discuss smoothing a noisy silhouette by well-known linear time-invariant filters. We also investigate the estimation of the slope and curvature of the original silhouette by pointwise estimates on the smoothed silhouette.

Given a sequence of points $\vec{x_i}$ on a silhouette, a sequence $\vec{x_i^*}$ corresponding to a smoothed silhouette is obtained by convolving the vector-valued sequence with a low-pass scalar impulse response h_i

$$\vec{x_i^*} = \sum_j h_j \vec{x_{i-j}} \quad \text{(A.11)}$$

It is desirable in general to use a symmetric (zero-phase) h_i so that a long straight line is not modified in the smoothing; the sum then extends from $-N$ to $+N$. In the case of an open contour, the output is not defined for the first and last N points. In the case of a closed contour, a circular convolution must be applied; this is obtained by considering the index $(i - j)$ modulo L, the length of the silhouette chain.

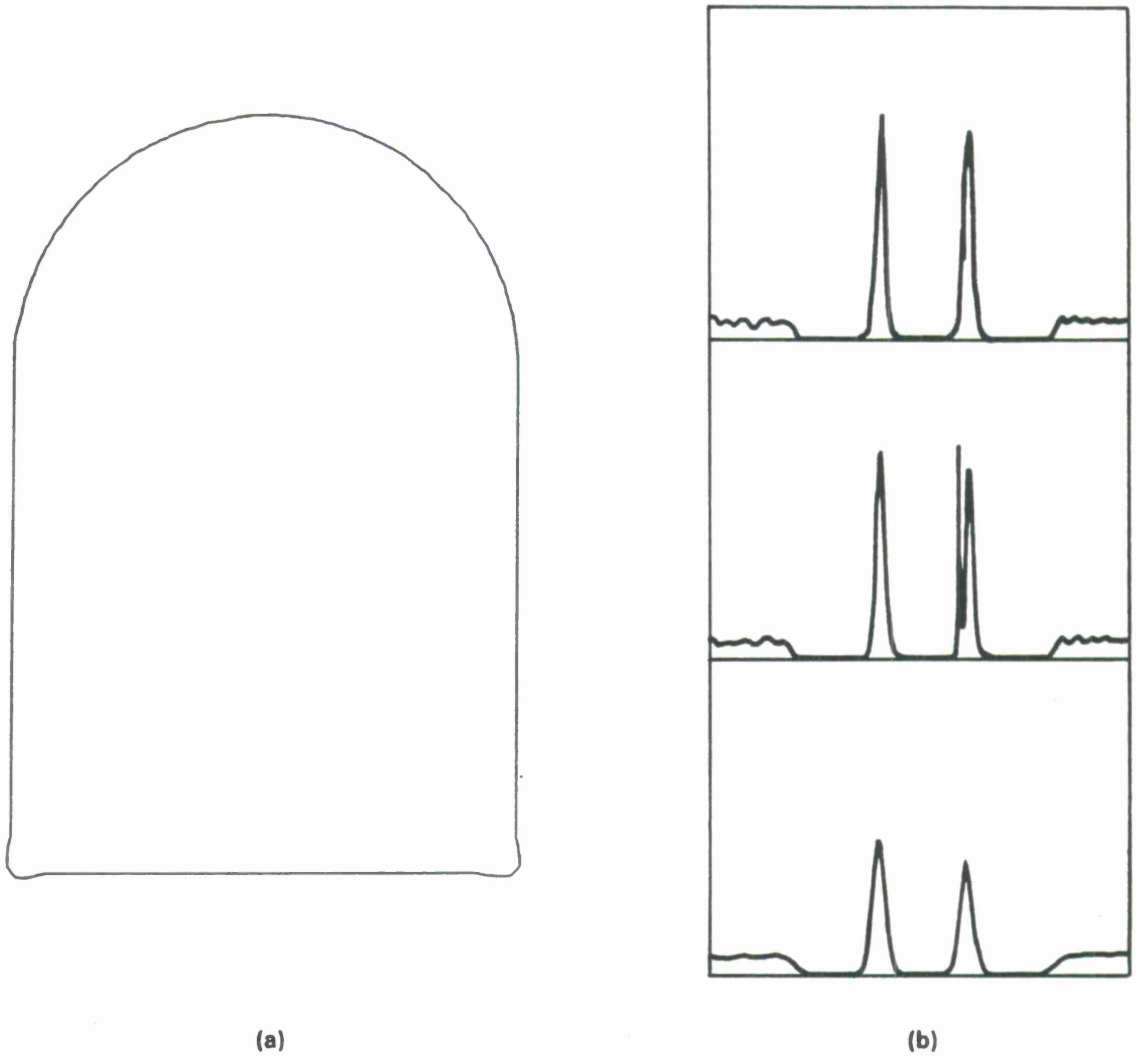


Figure A-8. Estimates of curvatures after smoothing with Gaussians of various widths: (a) original silhouette; (b) curvatures: sigma = 2, 3, 7.

$$\vec{x_i^*} = \sum_{j=-N}^{+N} h_j x_{[\vec{i-j}]_L} \tag{A.12}$$

The geometric interpretation of the above smoothing method is that each point is replaced by a weighted average of the $2N + 1$ points around itself, where the weights are given by the values of the sequence h_i. This interpretation shows that the weights in h_i should sum to 1. Although

the smoothing substantially reduces the amount of noise in the initial silhouette, it rounds the corners of polygonal silhouettes and reduces the radii of curvature on circular parts.

Decomposition of a Silhouette Based on Slopes and Curvatures

In this section, we discuss how estimates of local curvatures and slopes can be exploited to generate a decomposition of a silhouette in terms of straight edges. The process consists of first modeling long straight edges then smooth curves, each of which requires two steps. The first step of the strategy consists of deciding which points on the silhouette are likely to be on long straight edges. The detection of these candidates is based on estimates of curvatures and slopes at each point. The second step consists of grouping the candidate points into segments, and including additional points in the process. The third step consists of considering the points not included in the straight edges and detecting those points which are likely to be on smoothly curved parts of the silhouette. In the fourth step, the smooth candidates are grouped into smooth curve segments, potentially including more points, and these curve segments are each represented by a set of regularly spaced zero-length segments. The fifth and final stage consists of combining the straight segments and the infinitesimal segments representing the curves.

Selection of Points on Straight Edges In the absence of noise, a point is determined to be on a straight edge when the radius of curvature of the silhouette is zero at the point. In the presence of noise, a point is detemined to lie on a straight edge when the radius of curvature at the point is below some threshold. However, this finite threshold allows some other points to pass the test. To remove these spurious points, we use, in addition to the curvature, a criterion based on the smoothness of the silhouette at the point. An estimate of the ruggedness is given by the variation of the slope estimated after filtering with two different low-pass kernels. The ruggedness is low for points on straight lines and circles, and takes larger values near corners.

Grouping of Candidate Straight Edge Points The processing of candidate edge points must group points belonging to the same edge and must also retain the distinction between points on separate edges. The algorithm adopted in SILC accepts a string of consecutive points as a straight edge if it meets the following three criteria: (1) it contains a minimum fraction of selected points, (2) there are no excessive gaps in the sequence of selected points, and (3) the error of the linear fit to the points is lower than a threshold. The points are initially separated into likely groups by performing a simple Ramer decomposition of the silhouette with a moderate threshold. Only points in the same segment of the decomposition are considered for grouping.

Selection of Points on Smooth Curves A point is determined to be on a smooth curve if the absolute value of the radius of curvature estimated at the point is between two thresholds. The smoothness criterion described in the straight edge section is also applied here to eliminate spurious points retained by the above criterion. An additional smoothness criterion is used based

on the displacement of silhouette points by smoothing the curve with two kernels of different low-pass characteristics. The smoothness criterion accepts silhouette points only if the displacement is moderate.

Grouping of Candidate Smooth Curve Points The grouping of candidate curve points assembles points on the same silhouette curve while keeping points on distinct curves separate. Our algorithm accepts a group of consecutive points as a smooth curve region if it contains a minimum fraction of selected points, if there are no excessive gaps in the sequence of selected points, and if the curvature variation along the segment is less than a threshold. The last criterion avoids grouping curve points across inflections, for example.

Representing a Curve by Selected Points The SILC system performs recognition by matching straight edges in the image with straight edges in the object models. In order to accommodate both long straight edges and individual points on curves as input, a unique representation of edges was chosen. An edge is characterized by its center position, slope, and length, so that the representation is valid even when the length is zero. An individual curve segment is then represented by a series of such zero-length segments, evenly spread around the segment to best capture the shape of the curve.

Example An example of the results obtained with the algorithm described above is illustrated in Fig. A-9. The silhouette in (a) contains both straight edges and curved parts. The points circled in (b) are determined to correspond to a local flat shape, whereas the points circled in (c) correspond to curved segments. The resulting decomposition is displayed in (d).

A.2 SILHOUETTE PARSING STRATEGIES

As we mentioned earlier, the parsed silhouette must be a trade-off between an accurate description of each part of the silhouette that puts strong constraints on the object shape, and a conservative description that prevents recognition failure.

The choice of an appropriate parser depends on the type of object that the system is trying to detect in the image, and on the quality of the input data. We have observed that the Ramer decomposition is quite robust in the presence of noise but that it will not provide adequate representations of silhouette curves. With high resolution data, the method based on local characterizations of the silhouette shape provides adequate descriptions of both straight edges and curved parts of the silhouette. However, when this method is applied to very noisy silhouettes with both straight and curved parts, the decomposition has only a small number of edges and these model only a very small fraction of the original silhouette. As a consequence, recognition from these primitives is often ill-defined and results in excessive computational efforts. Although the Ramer approximation does not accurately model curved parts, we have observed that Ramer

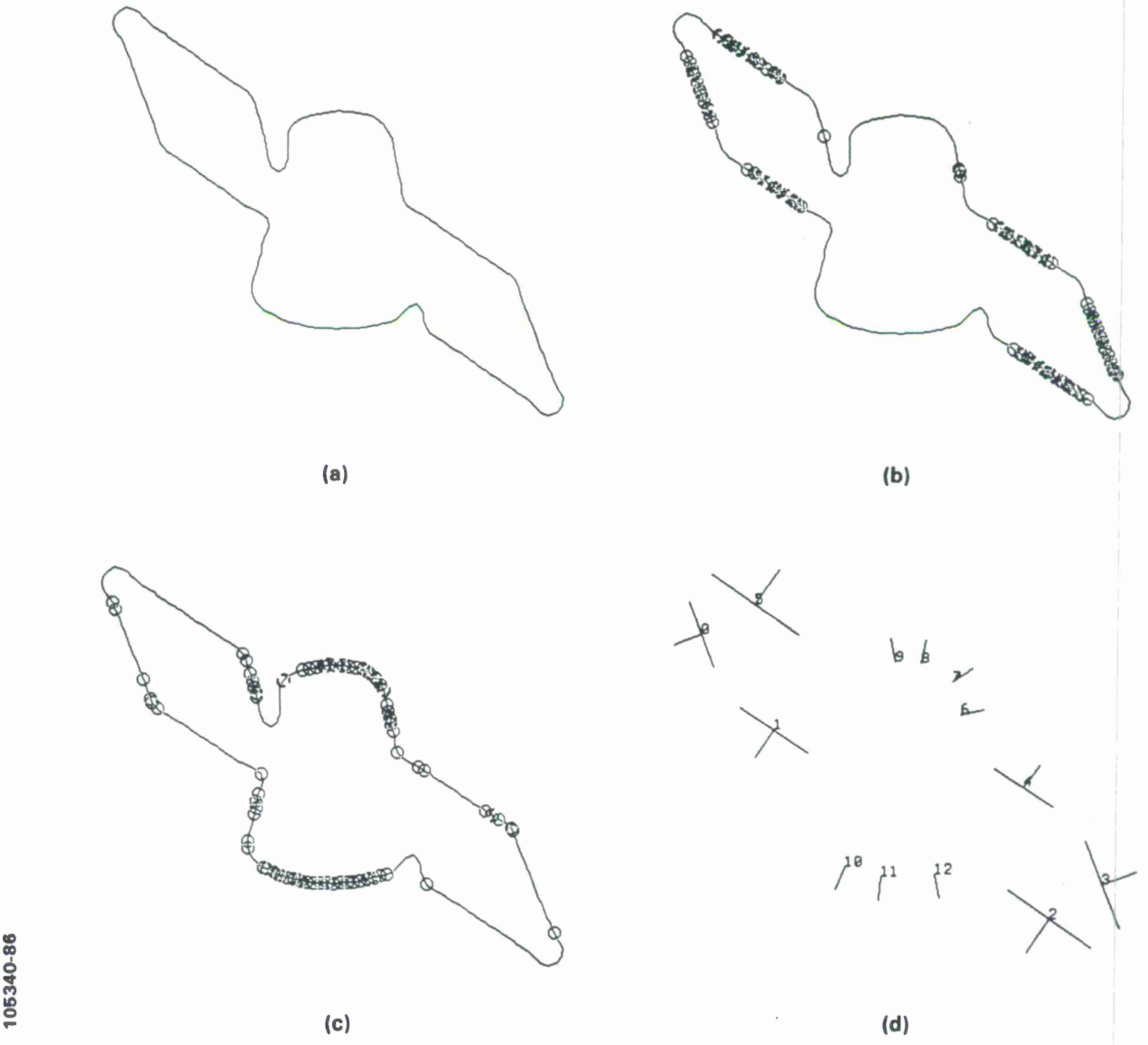


Figure A-9. Silhouette parsing based on curvatures: (a) original silhouette, (b) straight edge candidate points, (c) smooth curve candidate points, (d) parsed silhouette.

decompositions of noisy curved silhouettes usually result in better recognition than their counterparts based on local shape characteristics. However, the tolerance with respect to unmatched edges must be raised to model incorrect decompositions of the silhouette.

To conclude, we recommend the use of Ramer decompositions for polygonal silhouettes and for

very noisy silhouettes. Decompositions in terms of straight edges and curves, on the other hand are mostly appropriate for low noise silhouettes containing both straight and smoothly curved parts. Note that when the Ramer decomposition is used, the shorter segments of the decompositions must be discarded and the system parameters must be set up to tolerate a certain fraction of unmatched edges.

APPENDIX B
ESTIMATION OF THE IMAGING TRANSFORMATION

The verification of a silhouette interpretation, discussed in Section 7, consists of estimating an imaging transformation from the set of hypothesized edge pairings, constructing a synthetic silhouette of the model for this transformation, and comparing this synthetic silhouette with the observed silhouette. This appendix discusses the estimation of the viewing transformation given correspondences between image edges and model edges.

The estimation of an imaging transformation given matching features in the 2-D image and in the 3-D model space has been studied extensively, and solutions have been proposed for most types of projections in the case where the matching features are points. In SILC however, the pairings relate a silhouette edge element to a model edge. In the absence of occlusions and other artifacts, endpoints of the silhouette edge correspond to the endpoints of the related model edge and the algorithms developed for points can be exploited. However, in the presence of occlusions, and in the case of artifacts introduced by the front end processing, the correspondence between endpoints can rarely be guaranteed. Therefore, methods based specifically on correspondences of segments must be developed.

We first examine the constraints introduced by the correspondence between an image edge and a model edge, and estimate a minimum number of matches based on a count of the degrees of freedom. In that first section, we will also discuss representations of the orthographic transformation, with an emphasis on the explicit parameters of the transformation. We then characterize the recovery of the transformation as an error minimization problem and show that the equations are highly non-linear. Subsequently, we propose three strategies for solving the problem. The first is a closed-form least-squares suboptimal solution. It is applicable only when the problem is overconstrained, but it successfully exploits the redundant information to reduce noise effects when many edge matches are available. The second strategy is a closed form exact solution exploiting only a minimal number of edge matches. The third method is an iteration that converges towards the optimal solution of the problem. To conclude this appendix, we discuss how the three strategies are combined in SILC to determine appropriate solutions in most circumstances.

B.1 PROBLEM ANALYSIS

In this section, we consider the constraints introduced by the match of a silhouette edge and a model edge, estimate the number of degrees of freedom in an orthographic transformation, and then combine these to determine the minimum number of edge matches necessary to specify the transformation. Subsequently, we discuss representations for orthographic projections and finally set up the equations for the estimation of the projection.

B.1.1 Counting the Degrees of Freedom

When a 3-D model edge is matched with its exact projection on the image plane, this match sets four independent constraints on the viewing transformation, namely two for each endpoint. Unfortunately, the early processing of image data usually degrades image edges, by shortening edges at both ends and sometimes by breaking edges into two or more parts. In addition, a given model edge may appear only partially in the image in case of occlusions. As a consequence, only the following constraints can be exploited from the match of a 2-D edge and a 3-D edge. In the absence of noise, the orientation of the 2-D edge must fit the orientation of the projection of the 3-D edge and the lateral positions must also match exactly. However, the relative longitudinal position of the two edges is restricted only in the sense that the image edge must be included inside the projection of the model edge. Figure B-1 displays some possible combinations of an image edge and the projection of the corresponding model edge.

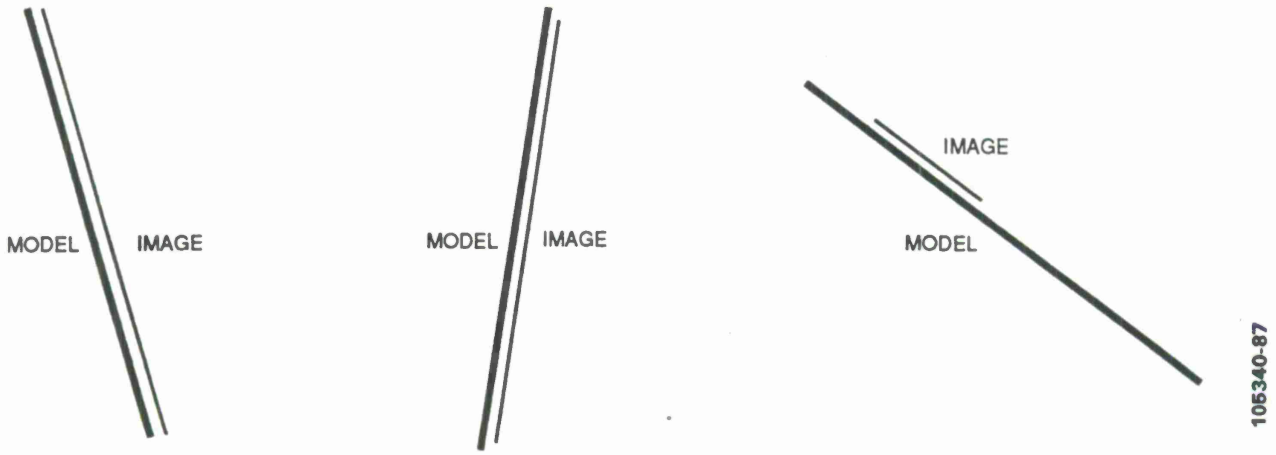


Figure B-1. Relative positions of an image edge and a projected model edge.

Among the constraints listed above, the orientation and the lateral position are equality constraints whereas the longitudinal position corresponds to two inequality constraints. In optimization problems, inequality constraints add useful information only when they are "active," that is when the solution constructed with the equality constraints violates only some of the inequalities. When estimating the imaging transformation, the inequalities are rarely active. We have therefore chosen to ignore them at this stage and to include the check of longitudinal positions in the verification procedure. The transformation is then estimated by requiring the match of orientation and lateral position between each image edge and the corresponding model edge; this is equivalent to considering edges as infinite lines. Each edge interpretation hence provides two independent equality constraints to the problem.

We now discuss the number of degrees of freedom (d.o.f.) in the imaging transformation. An orthographic projection is completely determined by the viewing direction (2 d.o.f.) and an orthographic transformation of the image plane (3 d.o.f.); it has therefore 5 degrees of freedom. The same answer is obtained by considering an orthographic projection as the composition of a general 3-D rotation (3 d.o.f.), a projection along a specified axis (0 d.o.f.), and a translation

in the image plane (2 d.o.f.). When the scale factor between the model and the image plane is unknown, this adds an extra degree of freedom to the transformation.

From the above analysis of constraints and degrees of freedom, it follows that three pairs of silhouette edges and model edges provide, in general, six constraints, which could be sufficient to characterize the orthographic projection, leaving one extra constraint when the scale is known. In some particular cases of interest, some of the constraints provided by silhouette edge measurements are redundant. For example, when two parallel 3-D model edges are related to the corresponding silhouette edges, the orientations of the silhouette edges are also parallel and only three constraints are provided by these two edges. Two colinear model edges provide only two constraints, and three parallel edges, three constraints. When the scale is known, sufficient constraints are provided by three edges if none of the model edges are colinear and if no more than 2 are parallel. When the scale factor is unknown, none of the model edges may be parallel or colinear.

B.1.2 Representations of Orthographic Projections

In this section, representations of orthographic projections are presented. The most natural representation of an orthographic projection for points characterized by Cartesian coordinates is a 2 x 4 matrix relating the 2-D coordinates $(x_\pi y_\pi)$ in the image plane to the 3-D coordinates (xyz) in model space by the following expression.

$$\begin{pmatrix} x_\pi \\ y_\pi \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{B.1}$$

In general, the above equation represents an affine projection which may include anisotropic scalings and shears. It represents an orthographic projection only when the following quadratic constraints are satisfied.

$$\begin{aligned} a_{11}^2 + a_{12}^2 + a_{13}^2 &= s \\ a_{21}^2 + a_{22}^2 + a_{23}^2 &= a_{11}^2 + a_{12}^2 + a_{13}^2 \\ a_{11}a_{21} + a_{12}a_{22} + a_{13}a_{23} &= 0 \end{aligned} \tag{B.2}$$

The scale factor s in the first equation may be set to its value when it is known; otherwise, only the last two equations provide constraints on the a_{ij}.

Note that in the above representation, the coefficients a_{14}, a_{24} represent the translation in the image plane and the other six coefficient represent a rotation in 3-D space. In fact, the coefficients a_{1j}, $i = 1,2,3$ are the components of the unit vector in 3-D model space along the image x-axis;

similarly, the coefficients a_{2j} correspond to the y-axis in the image plane. The constraints above are easily interpreted in terms of these vectors. Indeed, the first constraint specifies the length of the unit vector along the x-axis in the image. The second constraint specifies that both unit vectors in the image plane must have the same length. The third constraint requires that the two vectors be orthogonal.

A different representation of an orthographic projection is obtained by explicitly specifying a viewing direction and a transformation in the image plane. The projection is then characterized by a unit vector $(v_x v_y v_z)$ for the viewing direction, a scale factor s, the angle ψ and two translation components, t_x, t_y for the image plane transformation. The projection can be expressed in terms of these parameters as

$$\begin{pmatrix} x_\pi \\ y_\pi \end{pmatrix} = \begin{pmatrix} cos\psi & sin\psi \\ -sin\psi & cos\psi \end{pmatrix} \frac{s}{\sqrt{1-v_z^2}} \begin{pmatrix} -v_y & v_x & 0 \\ -v_x v_z & -v_y v_z & 1-v_z^2 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad \text{(B.3)}$$

The components v_x, v_y, v_z of the viewing direction vector must satisfy a quadratic constraint ensuring the unit norm of the vector.

The orthographic projection can also be represented by a 3-D rotation followed by the projection and an image translation. When the rotation is represented by a 3 x 3 matrix, a form equivalent to equation (B.1) is obtained. The rotation can also be represented by three independent Euler angles, by the four components of a unit quaternion, or by the exponential of an antisymmetric matrix.

B.1.3 Derivation of the Problem Equations

When the number of constraints exceeds the minimum number required, this redundancy can be exploited to reduce the effects of noise by formulating the estimation of the projection as an optimization problem, where the cost function is a weighted sum of squared errors between measured features and predicted features given the transformation, and where the optimization is carried over all possible transformations. The exact form of the equations depends on the representation chosen for orthographic projections. When the projection is represented by the Euler angles and by two image plane translations, the cost function is highly nonlinear but the 5 or 6 parameters of the transformation are independent. All other representations lead to constrained optimization problems. We have chosen to represent the projection by a 2 x 4 matrix; with this choice, the cost function is quadratic in the 8 parameters, and these parameters are constrained by 2 or 3 quadratic relations, depending on whether the scale is known or not. As we will see, the effect of measurement errors is easily expressed with these equations. We will first develop the 2 equality constraints corresponding to the match of one image edge with one model edge, then assemble these to construct the optimization function for a set of matched edges.

Constraint Equations for One Edge

The constraints relating the correspondence of a point $(x_s^1 y_s^1)$ in the image plane and a point $(x_m^1 y_m^1 z_m^1)$ in the model space, are given by

$$\begin{pmatrix} x_m^1 & y_m^1 & z_m^1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_m^1 & y_m^1 & z_m^1 & 1 \end{pmatrix} \begin{pmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{24} \end{pmatrix} = \begin{pmatrix} x_s^1 \\ y_s^1 \end{pmatrix} \tag{B.4}$$

The above equation is identical to (B.1), except that the coefficients a_{ij} have been structured as an 8-vector to emphasize that they are now considered as the unknowns.

As we mentioned earlier in this section, our system does not hypothesize correspondences between points, but between segments of the silhouette and those of the model. As reliable correspondences between endpoints of the segments cannot be made, the correspondences actually relate infinite lines in the model space to infinite lines in the silhouette plane. In both 2-D and 3-D space, an infinite line will be characterized by one point on the line, and a unit vector along the line. A 3-D line L_3 is hence characterized by the 3-D coordinates (x_{m0}, y_{m0}, z_{m0}) of one of its points, and by the 3 components (n_{mx}, n_{my}, n_{mz}) of its unit vector. The line L_3 is then the set of points

$$L_3 = \{(xyz) | x = x_{m0} + \lambda_1 n_{mx}, y = y_{m0} + \lambda_1 n_{my}, z = z_{m0} + \lambda_1 n_{mz}, \lambda_1 \in \Re\} \tag{B.5}$$

Similarly, a 2-D Line L_2 determined by a point (x_{s0}, y_{s0}) and by the unit vector (n_{sx}, n_{sy}) is the set of points

$$L_2 = \{(xy) | x = x_{s0} + \lambda_2 n_{sx}, y = y_{s0} + \lambda_2 n_{sy}, \lambda_2 \in \Re\} \tag{B.6}$$

When the line L_2 in the silhouette is associated with the line L_3 of the model, the following equations result.

$$\begin{pmatrix} x_{m0} + \lambda_1 n_{mx} & 0 \\ y_{m0} + \lambda_1 n_{my} & 0 \\ z_{m0} + \lambda_1 n_{mz} & 0 \\ 1 & 0 \\ 0 & x_{m0} + \lambda_1 n_{mx} \\ 0 & y_{m0} + \lambda_1 n_{my} \\ 0 & z_{m0} + \lambda_1 n_{mz} \\ 0 & 1 \end{pmatrix}^T \begin{pmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{24} \end{pmatrix} = \begin{pmatrix} x_{s0} + \lambda_2 n_{sx} \\ y_{s0} + \lambda_2 n_{sy} \end{pmatrix} \tag{B.7}$$

For each value of the parameter λ_1, there is one value of the parameter λ_2 for which the above equations are satisfied, and vice-versa. These equations can also be written as

$$\begin{bmatrix} x_{0s} + \lambda_2 n_{xs} = (a_{11}x_{0m} + a_{12}y_{0m} + a_{13}z_{0m} + a_{14}) + \lambda_1(a_{11}n_{xm} + a_{12}n_{xm} + a_{13}n_{xm}) \\ y_{0s} + \lambda_2 n_{ys} = (a_{21}x_{0m} + a_{22}y_{0m} + a_{23}z_{0m} + a_{24}) + \lambda_1(a_{21}n_{xm} + a_{22}n_{xm} + a_{23}n_{xm}) \end{bmatrix} \quad \text{(B.8)}$$

Elimination of the parameters λ_1, λ_2 provides the following two independent constraints.

$$\begin{pmatrix} n_{xm}n_{ys} & x_{0m}n_{ys} \\ n_{ym}n_{ys} & y_{0m}n_{ys} \\ n_{zm}n_{ys} & z_{0m}n_{ys} \\ 0 & n_{ys} \\ -n_{xm}n_{xs} & -x_{0m}n_{xs} \\ -n_{ym}n_{xs} & -y_{0m}n_{xs} \\ -n_{zm}n_{xs} & -z_{0m}n_{xs} \\ 0 & -n_{xs} \end{pmatrix}^T \begin{pmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{24} \end{pmatrix} = \begin{pmatrix} 0 \\ x_{0s}n_{ys} - y_{0s}n_{xs} \end{pmatrix} \quad \text{(B.9)}$$

In the above system, the first equation is equivalent to requiring the coefficients of λ_1 and λ_2 in (B.8) to be proportional, which is geometrically equivalent to requiring the projection of L_3 to be parallel to L_2. The second equation is obtained from (B.8) for $\lambda_1 = 0$, which is geometrically equivalent to requiring the projection of $(x_{0m}y_{0m}z_{0m})$ to be on L_2. The first equation hence expresses the constraint on orientation of the silhouette edge, whereas the second equation expresses the constraint on lateral position of the silhouette edge. A more natural expression of the second constraint would be obtained by requiring $(x_{0s}y_{0s})$ to be on the projection of L_3. Unfortunately, that constraint, obtained for $\lambda_2 = 0$ in (B.8), results in a quadratic equation for the a_{ij} given below. Due to its quadratic nature, this equation is more difficult to exploit; we have chosen to consider the second equation in (B.8) instead.

$$(a_{11}x_{0m} + a_{12}y_{0m} + a_{13}z_{0m} + a_{14} - x_{0s})(a_{21}n_{xm} + a_{22}n_{xm} + a_{23}n_{xm}) =$$

$$(a_{21}x_{0m} + a_{22}y_{0m} + a_{23}z_{0m} + a_{24} - y_{0s})(a_{11}n_{xm} + a_{12}n_{xm} + a_{13}n_{xm}) \quad \text{(B.10)}$$

Constraint Equations for N Edges

The equations corresponding to N matched edges are given by

$$\begin{pmatrix} n^1_{xm}n^1_{ys} & n^1_{ym}n^1_{ys} & n^1_{zm}n^1_{ys} & 0 & -n^1_{xm}n^1_{xs} & -n^1_{ym}n^1_{xs} & -n^1_{zm}n^1_{xs} & 0 \\ x^1_{0m}n^1_{ys} & y^1_{0m}n^1_{ys} & z^1_{0m}n^1_{ys} & n^1_{ys} & -x^1_{0m}n^1_{xs} & -y^1_{0m}n^1_{xs} & -z^1_{0m}n^1_{xs} & -n^1_{xs} \\ n^2_{xm}n^2_{ys} & n^2_{ym}n^2_{ys} & n^2_{zm}n^2_{ys} & 0 & -n^2_{xm}n^2_{xs} & -n^2_{ym}n^2_{xs} & -n^2_{zm}n^2_{xs} & 0 \\ x^2_{0m}n^2_{ys} & y^2_{0m}n^2_{ys} & z^2_{0m}n^2_{ys} & n^2_{ys} & -x^2_{0m}n^2_{xs} & -y^2_{0m}n^2_{xs} & -z^2_{0m}n^2_{xs} & -n^2_{xs} \\ & & & & \vdots & & & \\ n^N_{xm}n^N_{ys} & n^N_{ym}n^N_{ys} & n^N_{zm}n^N_{ys} & 0 & -n^N_{xm}n^N_{xs} & -n^N_{ym}n^N_{xs} & -n^N_{zm}n^N_{xs} & 0 \\ x^N_{0m}n^N_{ys} & y^N_{0m}n^N_{ys} & z^N_{0m}n^N_{ys} & n^N_{ys} & -x^N_{0m}n^N_{xs} & -y^N_{0m}n^N_{xs} & -z^N_{0m}n^N_{xs} & -n^N_{xs} \end{pmatrix} \begin{pmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{24} \end{pmatrix} =$$

$$= \begin{pmatrix} 0 \\ x^1_{0s}n^1_{ys} - y^1_{0s}n^1_{xs} \\ 0 \\ x^2_{0s}n^2_{ys} - y^2_{0s}n^2_{xs} \\ \vdots \\ 0 \\ x^N_{0s}n^N_{ys} - y^N_{0s}n^N_{xs} \end{pmatrix} \tag{B.11}$$

The above equations, together with the constraints in (B.2), specify the estimation problem. For practical values of N, the above system is in general overconstrained, and in the presence of noise in the measurements, is inconsistent. A classical solution consists in finding the set of unknowns for which the sum of squares of deviations from equality in the above equations is minimized. Rewriting the matrix equation (B.11) as

$$\mathbf{Ma} = \mathbf{b} \tag{B.12}$$

where $\mathbf{a}$ is the vector containing the unknowns. The least squares solution is obtained by minimizing the error (cost functional) E.

$$\min_{\mathbf{a}} E = \min_{\mathbf{a}} \left(\|\mathbf{Ma} - \mathbf{b}\|^2 \right) \tag{B.13}$$

Solution methods for the minimization problem defined by the above equation and by the constraints in (B.2) are developed in the next section.

B.2 SOLUTIONS OF THE ESTIMATION PROBLEM

There is no general closed form solution for the optimization problem described above, and optimum solutions can only be found iteratively. With iterative methods, issues of convergence to a fixed point and convergence to a local optimum arise. As the problem is highly nonlinear and has many variables, an algebraic approach to address these questions is difficult. We have therefore chosen to develop closed-form methods for finding a suboptimal transformation. This transformation can be used as the end result, or as a starting point for an iterative algorithm.

Two basic types of suboptimal solution methods will be developed. The first method starts by determining the redundant set of 8 parameters a_{ij} for the orthographic transformation, without enforcing the consistency constraints among these. The transformation itself is then determined by a valid set of parameters which closely approximates the inconsistent set. In the second type of method, only a small number of measurements (3 or 4 edges) is considered. The transformation is constructed geometrically from this limited set of measurements. Two examples of this strategy will be given.

The above methods for obtaining a suboptimal solution raise a number of issues. With the least squares method, an affine projection is first estimated. As this type of projection has eight degrees of freedom, its estimation from sparse data may be ill-conditioned or undefined, although the true transformation is not. On the other hand, the geometric methods use only a fraction of the available data and cannot reduce the errors by averaging over large sets of measurements. It turns out that, in general, the linear least squares method is most appropriate with highly redundant data, whereas the geometric methods are appropriate when only a few measurements on the object are available.

The two suboptimal methods described above are developed in the next two subsections. The results obtained with both methods can be improved upon by applying the iterative method developed in the third subsection.

B.2.1 Suboptimal Least-Squares Solution

A suboptimal solution of the optimization problem can be obtained by first solving for the 8 coefficients a_{ij} in (B.1) without considering the constraints in (B.2). It is apparent from (B.1) that the transformation equations are linear in these coefficients. Therefore, the least-squares cost function is quadratic in the unknowns and the solution to the suboptimal problem is linear. In general, the 8 coefficients obtained with the above method will not satisfy the constraints in (B.2). Geometric arguments are developed to infer a valid description of an orthographic projection from the 8 inconsistent coefficients.

Solving for 8 Independent Coefficients

The unconstrained least squares solution is obtained by minimizing the error

$$\min_{\mathbf{a}} E = \min_{\mathbf{a}} \left(\|\mathbf{M}\mathbf{a} - \mathbf{b}\|^2 \right) \tag{B.14}$$

The solution $\mathbf{a}^*$ of the above problem is also the solution of

$$\mathbf{M}^{\mathrm{T}}\mathbf{M}\mathbf{a}^* = \mathbf{M}^{\mathrm{T}}\mathbf{b}\mathbf{c} \tag{B.15}$$

which is given by

$$\mathbf{a}^* = \left(\mathrm{M}^{\mathrm{T}}\mathrm{M}\right)^{-1}\mathrm{M}^{\mathrm{T}}\mathbf{b} \tag{B.16}$$

The minimum error attained for the above solution is given by

$$E = \mathbf{b}^{\mathrm{T}}\mathbf{b} - \mathbf{b}^{\mathrm{T}}\mathrm{M}\left(\mathrm{M}^{\mathrm{T}}\mathrm{M}\right)^{-1}\mathrm{M}^{\mathrm{T}}\mathbf{b} = \mathbf{b}^{\mathrm{T}}\mathbf{b} - \mathbf{c}^{\mathrm{T}}\mathbf{a}^* \tag{B.17}$$

It has been observed experimentally in most common cases that the above method produces adequate estimates of the imaging projection. The method can be further refined by weighting the various equations in (B.11) according to estimates of corresponding measurement uncertainties.

Weighted Least Squares Estimation

The procedure described above minimizes the sum of squared deviations from equations such as (B.9) for each matching edge pair. Although reasonable transformations have been estimated with this method, it has two severe drawbacks. First, the relative importance of the orientation constraints and the position constraints cannot be controlled; this balance is actually affected by the scale of the measurements. Second, the relative importance of constraints arising from different edge pairs cannot be adjusted to reflect different levels of confidence in the estimated positions and orientations.

The above issues can be alleviated to some extent by weighting the deviation from each equation in (B.11). Our strategy is to exploit estimates of the measurement errors in silhouette edge position (x_{0s}, y_{0s}), and orientation (n_{xs}, y_{ys}), to weight each equation in (B.11), with an attempt to scale the worst-case deviation from each equation to 1.

Consider the first equation in (B.9). This equation is equivalent to

$$|\vec{n_{m\pi}}| \sin \phi_{m\pi s} = 0 \tag{B.18}$$

where $\vec{n_{m\pi}}$ is the model edge unit vector projected onto the image plane, and $\phi_{m\pi s}$ is the angle between this vector and the measured silhouette vector. If the measurement error on the silhouette edge orientation is $\Delta\phi$, the worst case analysis dictates a scale factor of $1/\sin\Delta\phi$ for this equation.

As discussed in Section B.1.3, the second equation in (B.9) expresses that the point (x_{0s}, y_{0s}) on the silhouette edge must be on the projection of the model edge. The deviation from this equation is the perpendicular distance between the observed silhouette points and the projected model edge. Deviations from this equation can be due both to the error in transversal silhouette edge position and to the silhouette edge orientation, when the silhouette point and the projected model point are not aligned. Specifically, the deviation is given by

$$\delta n_{0s} + l_{m\pi s} \sin \delta\phi_{0s} \tag{B.19}$$

where δn_{0s} is the error on transversal edge location, $l_{m\pi s}$ is the longitudinal distance between the projected model point and the silhouette point, and $\delta\phi_{0s}$ is the error on silhouette edge orientation. An upper bound on this error is given by

$$\Delta n_{0s} + l_m/2 \sin \Delta\phi_{0s} \tag{B.20}$$

where the Δ's represent error bounds, and l_m is the length of the model edge. The 1/2 factor in the above equations results from the assumption that the point (x_{0m}, y_{0m}, z_{0m}) is the mid-point of the model edge.

With the above weighting scheme, the solution is influenced more by the edges with the largest estimated accuracy, and, more importantly, is independent of measuring units. A substantial improvement in the accuracy of estimated imaging transformations has been noticed experimentally, when comparing weighted least squares estimated with their unweighted counterparts. An upper bound on the total error E for the weighted scheme is $2N$.

Limitations of the Method

Each combination of a model edge and a silhouette edge leads to two equations such as in (B.9). In order to solve for the 8 unknowns a_{ij}, four such combinations are sufficient, in the absence of particular alignments. However, as models are often idealized, and as man-made objects often contain large numbers of parallel edges, special cases can be more common than the general case. The effect of particular alignments is best understood by realizing that the transformation in (B.1) is an affine projection, in the absence of the constraints in (B.2). For an affine transformation, the numbers of constraints introduced by each measurement may be different from the ones derived in Section B.1.1. Indeed, since an affine transformation allows for shears, three parallel but non-planar model edges and their projections provide 4 constraints, as opposed to only 3 in the case of an orthographic transformation. A thorough analysis of the numbers of measurements required to solve for a unique affine transformation is beyond the scope of this report. A number of numerical experiments have been conducted and the conclusions derived from these are reported below.

It has been observed that four model edges with pairwise distinct orientations and no three orientations coplanar, combined with the corresponding image edges, are sufficient to determine the a_{ij}. In a number of cases of interest, a large number of model edges are oriented along three orthonormal directions. When only 3 orientations are possible for the 3-D edges, it is always necessary to use more than four edges to solve for the transformation. It has been experimentally observed that one 3-D edge parallel to, say the x-axis, and two pairs of 3-D edges parallel respectively to the y- and z-axes are generally sufficient to determine the transformation. Indeed, each pair of parallel edges specifies 3 constraints, and the edge along the x-axis specifies 2 constraints, for a total of 8 constraints. It is possible that for some lateral positions of these 5 edges in 3-D, the affine transformation would not be given uniquely by the correspondences between the 3-D edges and their projections, but this possibility has not been analyzed. Given an object with edges along three principal directions, the image may only contain edges along two of these directions in some cases, for example when the viewing direction is almost parallel to the third direction. It has been observed that three model edges in each of two orthogonal directions in 3-D are sufficient in general to solve for the affine transformation. In this case, the positions of the model edges in each triple may not be coplanar.

As with any numerical problem dealing with potentially noisy data, the systems of equations will rarely be degenerate. But when the underlying problem is degenerate, the equations will be ill-conditioned and produce irrelevant results. Furthermore, when the geometry of the data is close to one that produces a degenerate problem, numerical accuracies will be extremely poor.

Estimating an Orthographic Projection from 8 Coefficients

In the preceding text, a method was outlined for estimating the 8 coefficients of the imaging projection, given correspondences between silhouette edges and 3-D model edges. In the presence of noise, these 8 coefficients will not, in general, satisfy the constraints in (B.2). The issue is then to determine, from this set of 8 inconsistent coefficients, a set of 8 consistent coefficients which introduce the least amount of additional error between the projections of model features and silhouette features. Although this problem can be formulated analytically, a geometric solution is given here instead.

Among the 8 coefficients a_{ij} in (B.1), a_{14} and a_{24} represent the image translation t_x, t_y and are not affected by the consistency constraints. The other six coefficients can be cast into two 3-vectors, $\vec{a_1} = (a_{11}a_{12}a_{13})^T$ and $\vec{a_2} = (a_{21}a_{22}a_{23})^T$. As discussed in Section B.1.2, the problem can be phrased as recovering two orthonormal vectors $\vec{a_1^*}$ and $\vec{a_2^*}$ which must be as close as possible to the vectors $\vec{a_1}$ and $\vec{a_2}$ derived with the least-squares technique. Our choice, based on geometrical common sense, consists of choosing $\vec{a_1^*}$ and $\vec{a_2^*}$ as two orthonormal vectors in the plane of $\vec{a_1}$ and $\vec{a_2}$, with the bisectors of the two sets of vectors coinciding. After normalizing $\vec{a_1}$ and $\vec{a_2}$, the solution is given by

$$\left[\begin{array}{l} \vec{a_1^*} = \vec{a_1} + \frac{(1-|\vec{a_1} \times \vec{a_2}|)}{\vec{a_1} \cdot \vec{a_2}} \vec{a_2} \\ \vec{a_2^*} = \vec{a_2} + \frac{(1-|\vec{a_1} \times \vec{a_2}|)}{\vec{a_1} \cdot \vec{a_2}} \vec{a_1} \end{array} \right. \tag{B.21}$$

The above expressions are undefined for $\vec{a_1} \cdot \vec{a_2} = 0$, but in that case, the orthogonalization is unnecessary anyway! The vectors $\vec{a_1^*}$ and $\vec{a_2^*}$ must be normalized after having been evaluated with the above relations.

The viewing direction can be obtained directly from the inconsistent a_{ij}, as the vector product of $\vec{a_1}$ and $\vec{a_2}$.

$$\vec{v} = \frac{\vec{a_1} \times \vec{a_2}}{|\vec{a_1} \times \vec{a_2}|} \tag{B.22}$$

B.2.2 Geometric Solution Given Fewer than 8 Constraints

In the preceding sections, we showed that an orthographic transformation is determined by 5 independent constraints (6 when the scaling factor is unknown a-priori). We also discussed a strategy for estimating the transformation given correspondences between infinite lines in the scene and in the image. However, the method is applicable only when these correspondences imply at least 8 independent constraints on the transformation.

We have observed in our experiments that the data usually provides fewer than 8 constraints on the transformation only when all the matched edges are aligned with two principal orientations in the scene. We will therefore restrict our attention to such cases and, more specifically, to two particular situations. In the first case, two pairs of parallel edges in the scene (4 edges in all) are matched with their projections in the image. In the second case, two parallel lines and a third line with a different orientation in the scene (3 lines in all) are matched with their projections in the image. In both cases, we develop a method for estimating a viewing transformation with a fixed scale from the data. We do not attempt to exploit the redundant information to reduce the effects of biases and noise. Most situations which cannot be solved by the suboptimal least squares method can be reduced to one of the above cases. After discussing each of the two cases, we provide guidelines for selecting the 3 or 4 edges for the present method from a potentially larger pool of edges.

Our strategy for estimating the viewing direction is based on conceptually positioning the image plane as a projection plane in the scene, perpendicular to the viewing direction, in such a way that the projection of lines in the scene coincide with the matched lines in the image. The solution is then equivalent to finding the orientation of this plane in the scene. To determine the orientation of the projection plane with respect to the scene, we consider selected vectors in the image plane and estimate their orientation as vectors in the projection plane relative to the 3-D scene. It is sufficient to estimate the orientation of two such vectors in the projection plane to determine the viewing direction by their vector product. To fully determine the viewing transformation, it is then still necessary to determine the rotation and translation of the image plane axes with respect to the projection plane, but this is a much simpler problem.

In the discussion that follows, we first consider the constraints introduced by the correspondence of a pair of parallel image edges with a pair of parallel model edges and the representation of these constraints with vector algebra. We then discuss in detail the estimation of the viewpoint for the two cases mentioned above, including the selection of the best image edges to apply the estimation. Finally, we address the question of estimating the image plane transformation.

Vector Constraints from Correspondences of Edges

We will adopt the following notation for the various edges and vectors in the discussions to follow.

L_{s1}, L_{s2}	First pair of lines in the scene
L_{i1}, L_{i2}	Image lines matched to L_{s1}, L_{s2}
$\vec{s_1}$	Unit vector along L_{s1}, L_{s2}
$\vec{i_1}$	Unit vector along L_{i1}, L_{i2} in the image
$\vec{p_1}$	Unit vector along P_{s1}, P_{s2} in the projection plane
$\vec{s_{d1}}$	Unit vector along the distance from L_{s1} to L_{s2}
$\vec{i_{d1}}$	Unit vector along the distance from L_{i1} to L_{i2} in the image
$\vec{p_{d1}}$	Unit vector along the distance from L_{i1} to L_{i2} in the projection plane
d_{s1}	distance between L_{s1} and L_{s2} in the scene.
d_{i1}	distance between L_{i1} and L_{i2} in the image.
L_{s3}, L_{s4}	Second pair of lines in the scene
L_{i3}, L_{i4}	Image lines matched to L_{s3}, L_{s4}

In addition to the above vectors for the first pair of edges, the vectors $\vec{s_2}$, $\vec{i_2}$, $\vec{p_2}$, $\vec{s_{d2}}$, $\vec{i_{d2}}$, $\vec{p_{d2}}$ and the distances d_{s2}, d_{i2} denote the corresponding features for the second pair of parallel edges. Note that vectors such as $\vec{i_1}$ and $\vec{p_1}$ represent the same entity, except that the former is expressed in the the xy-axes of the image plane, whereas the latter is expressed in the xyz-axes of the scene, by conceptually positioning the vector in the projection plane within the scene.

A similar notation will be used for correspondences of 3 edges, except that $\vec{s_{d2}}$, d_{s2}, d_{i2} are not defined in this case, and that $\vec{i_{d2}}$ and $\vec{p_{d2}}$ are then simply normals to the third edge in the image and in the projection plane respectively.

The correspondences between L_{s1}, L_{s2}, L_{s3}, L_{s4} in the scene and L_{i1}, L_{i2}, L_{i3}, L_{i4} in the image imply the following constraints on the vectors defined above.

$$\vec{s_1} \cdot \vec{p_{d1}} = 0 \tag{B.23}$$

$$\vec{s_2} \cdot \vec{p_{d2}} = 0 \tag{B.24}$$

$$d_{s1}\,(\vec{s_{d1}} \cdot \vec{p_{d1}}) = d_{i1} \tag{B.25}$$

$$d_{s2}\,(\vec{s_{d2}} \cdot \vec{p_{d2}}) = d_{i2} \tag{B.26}$$

$$\vec{p_{d1}} \cdot \vec{p_{d2}} = \vec{i_{d1}} \cdot \vec{i_{d2}},\ \vec{p_{d1}} \cdot \vec{p_1} = 0,\ \vec{p_{d2}} \cdot \vec{p_2} = 0,\ \ldots \tag{B.27}$$

Among the above equations, (B.23) and (B.24) express the constraints that matched lines must have the correct orientation in the image, (B.25) and (B.26) express the consistency between distances of parallel edges in the scene and distances between their projections in the image, and (B.27) expresses that the edge vectors in the projection plane must have the same relative orientations as their counterparts in the image. In the case of correspondences between 3 lines in the scene and their projections in the image, only (B.23), (B.24), (B.25), and (B.27) apply. In the absence of external information, it is not possible to assign consistent signs to corresponding vectors such as $\vec{s_1}$ and $\vec{i_1}$. This is not an issue for equations (B.23), (B.24) and (B.27). Since L_{s1}

is matched to L_{i1} and L_{s2} to L_{i2} and not vice-versa, $\vec{s_{d1}}$ and $\vec{i_{d1}}$ can be assigned consistent signs, even though the latter is generally not the projection of the former. As a consequence, all the equations above are independent of arbitrary sign choices.

Solution Given Two Pairs of Parallel Edges

It is possible to solve equations (B.23) and (B.25) for $\vec{p_{d1}}$, since these are two independent constraints on a unit vector.

$$\vec{p_{d1}} = \frac{d_{i1}}{d_{s1}}\vec{s_{d1}} \pm \sqrt{1 - \frac{d_{i1}^2}{d_{s1}^2}}\,(\vec{s_{d1}} \times \vec{s_1}) \tag{B.28}$$

The above solution is easily verified by inspection. Similarly, (B.24) and (B.26) can be solved for $\vec{p_{d2}}$ as follows

$$\vec{p_{d2}} = \frac{d_{i2}}{d_{s2}}\vec{s_{d2}} \pm \sqrt{1 - \frac{d_{i2}^2}{d_{s2}^2}}\,(\vec{s_{d2}} \times \vec{s_2}) \tag{B.29}$$

The viewing direction can then be obtained as

$$\vec{v} \propto \pm\vec{p_{d1}} \times \vec{p_{d2}} \tag{B.30}$$

Since each of (B.28), (B.29), (B.30) admits two solutions, the above method produces 8 potential solutions for $\vec{v}$. However, we have not yet used the constraints on the consistency between vectors in the image and their counterparts in the projection plane, as expressed in (B.27). First, the sign in (B.30) can be determined by the sense of $\vec{i_{d1}}$ and $\vec{i_{d2}}$ (clockwise or counter-clockwise). Specifically,

$$sign\,(\vec{v} \cdot (\vec{p_{d1}} \times \vec{p_{d2}})) = sign\left(\vec{1_z} \cdot \left(\vec{i_{d1}} \times \vec{i_{d2}}\right)\right) \tag{B.31}$$

Second, the angle between the lines in the projection plane must match their angle in the image. Specifically,

$$\vec{p_{d1}} \cdot \vec{p_{d2}} = \vec{i_{d1}} \cdot \vec{i_{d2}} \tag{B.32}$$

In the absence of noise, the above equation generally reduces the number of solutions for $\vec{v}$ to two, which correspond to Necker cube reversals. In the presence of noise, the above equation will not be verified exactly, but only the solutions for $\vec{v}$ which produce small deviations from (B.32) should be retained.

Strategy for Choosing Two Pairs of Edges When the four edges exploited in the above method must be selected from a larger set of edges, it is important to choose the two pairs which provide the most accurate prediction of the viewing direction. This choice has two potential facets, namely the selection of a pair of edges among a set of parallel edges, and the selection of two such pairs among a set of pairs; we address the choices in this order.

Given a set of parallel edges, the method described above estimates a normal $\vec{p_{d1}}$ to these edges in the projection plane. The accuracy of the estimate of this vector with (B.28) should dictate the choice of the best pair. We assume that $\vec{s_1}$, $\vec{s_{d1}}$, which are vectors in the scene, are known exactly from the object models. The distance d_{s1} is also assumed error free, but d_{i1} may be affected by image measurements; only those two distances vary with the choice of the pair of edges considered. The error on $\vec{p_{d1}}$ can be estimated as

$$|\Delta \vec{p_{d1}}| = \frac{\Delta d_{i1}}{\sqrt{d_{s1}^2 - d_{i1}^2}} \tag{B.33}$$

The error estimated by the above formula is minimized for edges which are accurately located in the image (small Δd_{i1}), which are distant in the model (large d_{s1}), and close in the image (small d_{i1}). The above criterion must be applied to choose the best two edges among each set of parallel model edges matched to image edges. When more than two sets of parallel edges are available to estimate the viewing direction, the choice of two of these must be dictated by the accuracy on the estimate of $\vec{v}$ with (B.30). Note that the unit vector $\vec{v}$ is obtained by normalizing the right side of (B.30). Therefore, the error on $\vec{v}$ can be expressed as

$$|\Delta \vec{v}| = \frac{1}{|\vec{p_{d1}} \times \vec{p_{d2}}|} |\Delta(\vec{p_{d1}} \times \vec{p_{d2}})_{\perp \vec{v}}| \tag{B.34}$$

In the above equation, the second factor of the right side is the error of the cross product in (B.30), projected orthogonally to $\vec{v}$; it is a quite complex function of all the vectors involved, and depends ultimately on the accuracy of both $\vec{p_{d1}}$ and $\vec{p_{d2}}$. The first factor reveals that errors will be magnified by a potentially large factor when the vectors $\vec{p_{d1}}$ and $\vec{p_{d2}}$ are nearly aligned. We have decided to base the selection of two pairs of vectors on the minimization of

$$(|\Delta \vec{p_{d1}}| + |\Delta \vec{p_{d2}}|)/(\vec{p_{d1}} \times \vec{p_{d2}}) \tag{B.35}$$

This choice will favor pairs which provide an accurate estimate of $\vec{p_d}$ and a couple of pairs which represent sufficiently different orientations in the image plane. In the above discussion, we did not consider the estimation of $\vec{i_1}$, $\vec{i_{d1}}$, d_{i1} themselves. When a pair of parallel edges is selected from a larger set of parallel edges, $\vec{i_1}$ should be estimated as a weighted average of the orientations of all the edges in the set; then, $\vec{i_{d1}}$ is chosen normal to $\vec{i_1}$, and d_{i1} is measured along $\vec{i_{d1}}$.

Solution Given Two Parallel Edges and a Third Edge

We consider now the case where the parallel lines L_{s1}, L_{s2} and the line L_{s3} in the scene are matched to L_{i1}, L_{i2}, L_{i3} in the image. In this case again, the vector $\vec{p_{d1}}$ in the projection plane can be estimated with equation (B.28). However, a different method must be devised to determine a second vector in the projection plane. Since (B.24) is an explicit constraint on $\vec{p_{d2}}$, we choose to estimate the orientation of this particular vector in the projection plane. Since $\vec{p_{d2}}$ is a unit vector, the two constraints (B.24) and $\vec{p_{d1}} \cdot \vec{p_{d2}} = \vec{i_{d1}} \cdot \vec{i_{d2}}$ in (B.27) are sufficient to narrow the choices for $\vec{p_{d2}}$ down to two solutions. Indeed, the problem is to find a vector given its length and the scalar product with two vectors; the solution can be found, for example in [13], page 457-458. The particular solution in our case is given by

$$\vec{p_{d2}} = \frac{\vec{i_{d1}} \cdot \vec{i_{d2}}}{1 - (\vec{s_2} \cdot \vec{p_{d1}})} (\vec{p_{d1}} - (\vec{s_2} \cdot \vec{p_{d1}})\vec{s_2}) \pm \lambda (\vec{s_2} \times \vec{p_{d1}}) \tag{B.36}$$

where the parameter λ is determined by adjusting the length of $\vec{p_{d2}}$ to 1. The above solution is easily verified by inspection. After estimating $\vec{p_{d2}}$ with equation (B.36), the viewing direction is again estimated with equation (B.30). Among the 8 solutions that are obtained, half can be eliminated by the constraint on the relative orientation of $\vec{p_{d1}}$ and $\vec{p_{d2}}$ in (B.31). Some of the solutions of (B.36) correspond to a complex value for the parameter λ and can be further eliminated. Depending on the orientations of edges, the above method provides 2 or 4 valid solution to the estimation problem.

Choosing 3 Edges for Estimating the Viewing Direction When estimating the viewing direction with three edges, two of which are parallel, it is important to properly choose the edges to which the above method is applied, when this choice is possible. We analyze here the error on the viewing vector resulting from inaccuracies of the image data, and develop a strategy for choosing the appropriate triple of edges based on this analysis.

When more than two image edges are matched to parallel edges in the scene, the selection of the best pair of edges should proceed along the lines of paragraph B.2.2. The errors related to the third edge are now addressed. Errors in the estimate of $\vec{p_{d2}}$ with (B.36) arise from inaccuracies in $\vec{i_{d1}} \cdot \vec{i_{d2}}$ and in $\vec{p_{d1}}$. The correct expression for the errors is quite complex, but in general, the estimate is well conditioned when $|\vec{s_2} \times \vec{p_{d1}}|$ and $\vec{i_{d1}} \times \vec{i_{d2}}$ are large. Given a set of parallel edges, we choose the third edge by minimizing the product

$$(\vec{s_2} \cdot \vec{p_{d1}}) \left(\vec{i_{d1}} \cdot \vec{i_{d2}}\right) \tag{B.37}$$

Estimation of the Image Plane Rotation and Translation

In the above paragraphs, we developed a method for estimating the viewing direction of a fixed scale orthographic transformation from correspondences of 3 or 4 edges in the scene to their counterparts in the image. In order to fully determine the viewing transformation, however, it

is also necessary to estimate the rotation and translation inside the image plane. We develop here a method for performing these estimates. The method is based on choosing an arbitrary rotation and translation for the projection plane axes and synthesizing the projection of L_{s1}, L_{s2}, ... in that plane. The comparison of these projections with the corresponding lines L_{i1}, L_{i2}, ... in the image provides the discrepancies in position and orientation; the parameters of the correct transformation are then adjusted to compensate for these discrepancies.

The discrepancy between the orientations of the synthesized projections and those of the lines in the image can be obtained easily as a weighted average of the angle differences between the image lines and the projected lines. However, it is a little more intricate to characterize the relative translational position of the two sets of edges, since the endpoints of corresponding edges are not assumed to match. In order to characterize the position of a set of infinite lines in the planes, we have developed the concept of a center of mass of a set of lines. This point is characteristic of a set of lines in 2-D or in 3-D, and is unaffected by translations and/or rotations. The translation in the image plane is estimated by computing the center of mass of the image lines and the center of mass of the projected model lines in the image plane. The image plane translation is then simply the vector distance between these two points. In order to take into account varying degrees of accuracy on the position and orientation of the edges in the image, corresponding weights are set on each edge for the estimation of the centers of mass.

Estimation of the Center of Mass of a Set of Weighted Infinite Lines We define the center of mass $G_L(x_G, y_G)$ of a set of lines L_i as the point which minimizes the sum of squared distances to each L_i. The center of mass of a weighted set of lines is obtained by multiplying each distance by the weight W_i associated with the corresponding line. Let each line L_i be determined by a unit vector $\vec{u_i}$ along the line and the coordinate vector $\vec{x_i}$ of a point on the line. The center of mass we are trying to determine is characterized by its coordinate vector $\vec{x_G}$; it is determined by minimizing the weighted sum

$$E = \sum W_i d^2(L_i, G_L) = \sum W_i \left[\|\vec{x_i} - \vec{x_G}\|^2 - [(\vec{x_i} - \vec{x_G}) \cdot \vec{u_i}]^2 \right] \tag{B.38}$$

The above weighted distance measure is minimized when $\vec{x_G}$ solves

$$\sum \left[W_i (I - \vec{u_i}\vec{u_i}^T)\vec{x_i} \right] = \left[\sum W_i (I - \vec{u_i}\vec{u_i}^T) \right] \vec{x_G} \tag{B.39}$$

Note that under this definition, the "center of mass" of a single line or a set of parallel lines is not a point but a line. However, in the general case, this definition yields the intuitive location of the center of mass.

It is easy to verify that the preceding equations for $\vec{x_G}$ are invariant with respect to translations and rotations. However, the "center of mass" defined by the above method is not conserved in projections, or otherwise said, the center of mass of a set of lines in the image plane is not the projection of the center of mass of the corresponding lines in the scene. A simple counterexample is found by considering two skewed infinite lines in 3-D and their projection on a 2-D plane. The center of mass in 3-D is the midpoint of the perpendicular to the two lines, whereas the 2-D

center of mass is simply the intersection point of the two lines. Generally, the midpoint of the perpendicular to two lines does not project onto the intersection of the projection of the two lines in 2-D.

This lack of consistency between centers of mass in 2-D and in 3-D was cited only for the sake of completeness; it is not an issue for our application since our comparison is only based on the center of mass of lines in the image and that of the projection of the scene lines in the projection plane.

B.2.3 Iterative Estimation of the Viewing Direction

In the preceding sections of this section, several closed-form methods were discussed for estimating the imaging transformation given a set of corresponding infinite lines. However, all these methods are suboptimal in nature; optimal solutions can be obtained only with iterative methods. We discuss here the iterative update of a viewing transformation, given correspondences of straight lines. This iteration can be used to refine a transformation estimate obtained by one of the methods discussed above. Although this iteration could be used to solve the estimation problem by itself, we have experienced divergence of the iteration when the starting point is remote from the solution point. Consequently, it is advisable to use one of the closed-form solutions methods to produce a first estimate of the transformation, then to use the iteration to refine the transformation.

In this context, we will adopt the following notation for the transformation

$$\vec{x_2} = I_{23} A \vec{x_3} + \vec{a_T} \text{ , with } A^T A = s^2 I \tag{B.40}$$

In the above equation, A is a 3 x 3 orthogonal matrix with scale s, I_{23} is the projection matrix which retains the first two rows of A, and $\vec{a_T} = (a_{14}\ a_{24})^T$ is the 2-vector representing the translation in the image plane. During the iterative update of A, an estimate $A^{(N+1)}$ is evaluated from the estimate $A^{(N)}$ by applying small rotations and a small change of scale as in

$$A^{(N+1)} = (1 + \Delta s) R_x(\Delta\phi_x) R_y(\Delta\phi_y) R_z(\Delta\phi_z) A^{(N)} \tag{B.41}$$

Where the rotation matrices R_x, R_y, R_z are given by

$$R_x(\Delta\phi_x) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -\Delta\phi_x \\ 0 & \Delta\phi_x & 1 \end{pmatrix}, \quad R_x(\Delta\phi_y) = \begin{pmatrix} 1 & 0 & -\Delta\phi_y \\ 0 & 1 & 0 \\ \Delta\phi_y & 0 & 1 \end{pmatrix},$$

$$R_x(\Delta\phi_z) = \begin{pmatrix} 1 & -\Delta\phi_z & 0 \\ \Delta\phi_z & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{B.42}$$

The combination of these three rotations is, for small rotation angles,

$$R_x R_y R_z = \begin{pmatrix} 1 & -\Delta\phi_z & -\Delta\phi_y \\ \Delta\phi_z & 1 & -\Delta\phi_x \\ \Delta\phi_y & \Delta\phi_x & 1 \end{pmatrix} \tag{B.43}$$

The update of the translation vector is written as

$$a_T^{(N+1)} = a_T^{(N)} + \Delta a_T \tag{B.44}$$

In Section B.1.3, we determined two constraints for each match of a scene line with an image line (B.9). Both constraints are linear in the unknown coefficients of the transformation and can be written formally as

$$\alpha_1 a_{11} + \alpha_2 a_{12} + \alpha_3 a_{13} + \alpha_4 a_{21} + \alpha_5 a_{22} + \alpha_6 a_{23} + \alpha_7 a_{14} + \alpha_8 a_{24} = \alpha_9 \tag{B.45}$$

The updates of A and of a_T from one step of the iteration to the next are determined by applying the measurement constraints such as above to the next estimates $A^{(N+1)}$ and $a_T^{(N+1)}$. These result in equations which can be solved for the update parameters Δs, $\Delta\phi_x$, $\Delta\phi_y$, $\Delta\phi_z$, $\Delta\vec{a_T}$. The constraint on these increments corresponding to the constraint on the a_{ij}'s given above can be expressed as

$$\begin{aligned} &- (\alpha_4 a_{31} + \alpha_5 a_{32} + \alpha_6 a_{33})\,\Delta\phi_x \\ &- (\alpha_1 a_{31} + \alpha_2 a_{32} + \alpha_3 a_{33})\,\Delta\phi_y \\ &+ (-\alpha_1 a_{21} - \alpha_2 a_{22} - \alpha_3 a_{23} + \alpha_4 a_{11} + \alpha_5 a_{12} + \alpha_6 a_{13})\,\Delta\phi_z \\ &+ (\alpha_1 a_{11} + \alpha_2 a_{12} + \alpha_3 a_{13} + \alpha_4 a_{21} + \alpha_5 a_{22} + \alpha_6 a_{23})\,\Delta s \\ &+ \alpha_7 \Delta a_{14} \\ &+ \alpha_8 \Delta a_{24} \\ &= \alpha_9 - (\alpha_1 a_{11} + \alpha_2 a_{12} + \alpha_3 a_{13} + \alpha_4 a_{21} + \alpha_5 a_{22} + \alpha_6 a_{23} + \alpha_7 a_{14} + \alpha_8 a_{24}) \end{aligned}$$

The suboptimal closed-form solution of the viewing transformation developed in Section B.2.1 consisted of solving a set of equations such as (B.45) for the eight unknowns a_{ij}; the iterative method consists of updating an initial estimate of the transformation by solving a set of equations such as (B.46) for the five or six unknown Δs, $\Delta\phi_x$, $\Delta\phi_y$, $\Delta\phi_z$, Δa_{14}, Δa_{24}, with Δs fixed to 0 in the case of a fixed scale transformation. In both cases, the sets of equations are solved by weighted linear least squares.

B.3 SOLVING FOR THE TRANSFORMATION IN SILC

In SILC, the estimation of the transformation combines the three methods proposed above to solve the estimation problem in most common cases. First, a solution is attempted by the underconstrained least-squares method. To this end, the matrix $\mathbf{M}^T\mathbf{M}$ in (B.15) is evaluated, and an estimate of the condition number of this matrix is used as an indicator of the applicability of this method. In the negative case, sets of model and image edges are selected to apply the vector method developed in Section B.2.2. Whether the solution is obtained initially with the least squares method of with the vector method, an improved final solution is obtained by applying the iterative update described in Section B.2.3, nominally for three iterations. We have experimentally observed that the above strategy produces good results in most practical situations.

ACKNOWLEDGEMENTS

We would like to thank Dan Dudgeon and Bob Gabel for their support and guidance. They were a constant source of ideas and motivation, and together they proof read the mounds of paper that constituted all of the drafts.

Special thanks go to Glenn Adams, who solved many problems arising in the software environment and helped to keep the system running. We are grateful to all the members of the Machine Intelligence Technology group for creating a friendly and stimulating work environment.

This work was performed under the support of the Lincoln Laboratory Innovative Research Program.

REFERENCES

1. H. Asada and M. Brady, *The Curvature Primal Sketch*, Memo AIM-758, M.I.T. Artificial Intelligence Laboratory, 1984.

2. H.G. Barrow and J.M. Tennenbaum, Interpreting Line Drawings as Three Dimensional Surfaces, *Artificial Intelligence*, 17:75–116, 1981.

3. B. Bhanu and O.D. Faugeras, Shape Matching of Two-Dimensonal Objects, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(2):137–156, 1984.

4. R.C. Bolles, P. Horaud, and M.J. Hannah, 3DPO: A Three-Dimensional Part Orientation System, In *Proceedings of the Eight International Joint Conference on Artificial Intelligence*, pages 1116–1120, Karlsruhe, W.Germany, August 8-12 1983.

5. R. A. Brooks, Symbolic Reasoning Among 3-D Models and 2-D Images, *Artificial Intelligence*, 17:285–348, 81.

6. I. Chakravarty and H. Freeman, Characteristic Views as a Basis for Three-Dimensional Object Recognition, In *Proceedings of The Society for Photo-Optical Instrumentation Engineers Conference on Robot Vision*, pages 37–45, SPIE, Arlington, Va., May 6-7 1982.

7. J. G. Dunham, Optimum Piecewise Linear Approximation of Planar Curves, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8(No. 1):67–75, January 1986.

8. H. Freeman, Techniques for the Digital Computer Analysis of Chain-encoded Arbitrary Plane Curves, In *Proceedings of the National Electronics Conference*, pages 431–432, 1961.

9. R. R. Goldberg and D. G. Lowe, Verification of 3-D Parametric Models in 2-D Image Data, In *IEEE Computer Society Workshop on Computer Vision*, Miami Beach, FL, Nov. 30 - Dec. 2 1987.

10. W. E. L. Grimson, The Combinatorics of Local Constraints in Model-Based Recognition and Localization from Sparse Data, *Journal of the ACM*, 33(4):658–686, October 1986.

11. W. E. L. Grimson and T. Lozano-Perez, Model-Based Recognition and Localization from Sparse Range or Tactile data, *International Journal of Robotics Research*, 3(3):3–35, Fall 1984.

12. A. B. Gschwendtner, R. C. Harney, and R. J. Hull, Coherent IR Radar Technology, In D. K. Killinger and A. Mooradian, editors, *Optical and Laser Remote Sensing*, pages 327–340, Springer-Verlag, 1983.

13. B. K. P. Horn, *Robot Vision*, MIT Press, Cambridge, MA, 1986.

14. M. K. Hu, Visual Pattern Recognition by Moment Invariants, *IRE Transactions on Information Theory*, Vol. 8:179–187, 1962.

15. K. Ikeuchi, Precompiling a Geometric Model into an Interpretation Tree for Object Recogintion in Bin Picking, In *Proceedings of the Image Understanding Workshop*, pages 321–339, DARPA, Los Angeles, CA, February 23-25 1987.

16. J. J. Koenderink and A. J. Van Doorn, Internal Representation of Solid Shape with Respect to Vision, *Biological Cybernetics*, Vol. 32(No. 4):211–216, 1979.

17. Y. Kurozumi and W. A. Davis, Polygonal Approximation by the Minimax Method, *Computer Graphics and Image Processing*, Vol. 19(No. 2):248–264, 1982.

18. D. G. Lowe, Three-Dimensional Object Recognition from Single Two-Dimensional Images, *Artificial Intelligence*, 31(3):335–395, March 1987.

19. D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*, Freeman, San Fransisco, CA, 1982.

20. U. Ramer, An Iterative Procedure for the Polygonal Approximation of Planar Curves, *Computer Graphics and Image Processing*, Vol. 1(No. 2):244–256, 1972.

21. L. G. Roberts, *Machine Preception of Three-Dimensional Solids*, Technical Report TR-315, M.I.T. Lincoln Laboratory, May 1963.

22. P. G. Selfridge, *Reasnoning about Success and Failure in Aerial Image Understanding*, PhD thesis, Dept. of Computer Science, University of Rochester, Rochester, NY, 1981.

23. J. Sklansky and G. A. Davison, Recognizing Three-Dimensional Objects by their Silhouettes, *Journal of the SPIE*, 10:10–17, Nov./Dec. 1971.

24. D. W. Thompson and J. L. Mundy, Three-Dimensional Model Matching from an Unconstrained Viewpoint, In *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, pages 208–220, Raleigh, NC, March 31 - April 3 1987.

25. P. Van Hove, *Silhouette-Slice Theorems*, PhD thesis, M.I.T., Department of Electrical Engineering and Computer Science, Cambridge, MA, September 1986.

26. P. Van Hove, *Silhouette-Slice Theorems*, Technical Report TR-764, M.I.T. Lincoln Laboratory, Lexington, MA, March 1987.

27. J. G. Verly, T. R. Esselman, and D. E. Dudgeon, Silhouette Feature Extraction in Laser-Radar Range Imagery, In *Proceedings of the IRIS Active Systems Specialty Group Meeting*, November 1986.

28. J. G. Verly, P. L. Van Hove, R. L. Walton, and D. E. Dudgeon, Silhouette Understanding System, In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Tokyo, Japan, April 7-11 1986.

29. A. Weiss and H. Nawab, A Representation for the Orientation-Dependent Appearance of 3-D Objects, In *Proceedings of the the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New York, NY, April 1988.

30. C. T. Zahn and R. Z. Roskies, Fourier Descriptors for Plane Closed Curves, *IEEE Transactions on Computers*, C-21(3):269–281, March 1972.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release; distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
Technical Report 817

5. MONITORING ORGANIZATION REPORT NUMBER(S)
ESD-TR-88-194

6a. NAME OF PERFORMING ORGANIZATION
Lincoln Laboratory, MIT

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION
Electronic Systems Division

6c. ADDRESS *(City, State, and Zip Code)*
P.O. Box 73
Lexington, MA 02173-0073

7b. ADDRESS *(City, State, and Zip Code)*
Hanscom AFB, MA 01731

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
Air Force Systems Command, USAF

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and Zip Code)*
Andrews AFB
Washington, DC 20334

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
	1		

11. TITLE *(Include Security Classification)*
Model-Based Silhouette Recognition

12. PERSONAL AUTHOR(S)
Patrick L. Van Hove and Richard A. Jaenicke

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT *(Year, Month, Day)*	15. PAGE COUNT
Technical Report	FROM _____ TO _____	25 November 1988	160

16. SUPPLEMENTARY NOTATION
None

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*

computer vision
scene understanding
object recognition
silhouette
constrained search
constraint propagation
perceptual reasoning
combinational complexity
three dimensional objects

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

We present a system for recognizing 3-D objects at unknown orientations from their 2-D silhouettes. The geometric description of an object model is provided in CAD form and is then compiled into a set of geometric constraints for a large set of viewing directions. The silhouette is parsed into a set of straight edges, and these edges are compared to the edge of the model by conceptually structuring all possible interpretations in a tree. This enormous search space is pruned by extending the interpretation tree search of Grimson and Lozano-Perez to work for the 3-D model/2-D data case. This includes a precise analysis of the propagation of errors in the position and orientation of silhouette edges, which then provide adequate constraints for pruning the search tree. Any hypotheses that survive the pairwise constraints of tree search are verified by synthesizing a silhouette of the model for the hypothesized orientation and comparing this synthetic silhouette to the observed silhouette.

Based only on silhouette data, the system can find all plausible interpretations of the data, including symmetric viewpoints. The system performs in the presence of unknown viewpoint, moderate scale uncertainties, occluding objects and degradations in the silhouette shape due to image noise and image processing artifacts. These characteristics should enable the system to perform well in applications where images have reasonable spatial resolution but were limited resolution in the signal (intensity or range) reduces the information in the data to a silhouette.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
Lt. Col. Hugh L. Southall, USAF

22b. TELEPHONE *(Include Area Code)*
(617) 981-2330

22c. OFFICE SYMBOL
ESD/TML