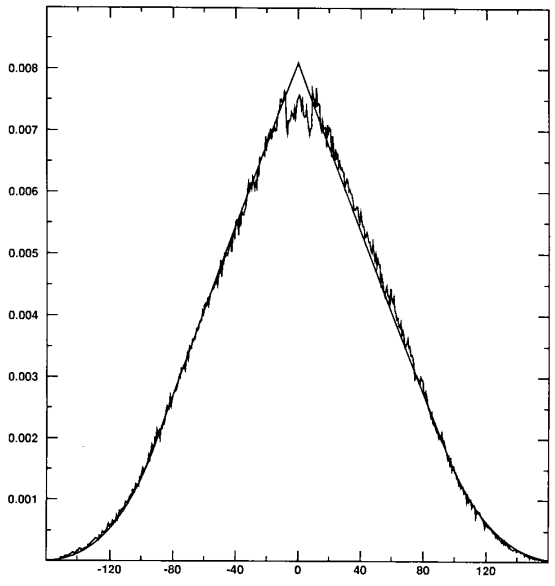


Fig. 2. Histogram and p.d.f. of Δz.

VI. Experimental Results

A stereo imaging system was simulated in order to assess the accuracy of the derived probability density function and the expected value of the range error. Typical values were chosen for the various system parameters (b = 973 m, f = 30 mm, $z_{\min}$ = 4456 m, $z_{\max}$ = 5922 m, and δ = 137 μm). A pseudorandom number generator was used to generate the x and z coordinates of an object point lying within the field of view. Perspective projection, in accordance with (1) and (2), was used to obtain the left and right image plane coordinates corresponding to this object point. Next, these x_L and x_R coordinates were rounded off to the nearest pixel, yielding $\hat{x}_L$ and $\hat{x}_R$. Finally, the measured disparity, $\hat{d} = \hat{x}_L - \hat{x}_R$, was used to estimate the z coordinate, via the inverse perspective projection equation, $\hat{z} = bf/\hat{d}$. The range estimation error was then given by $\Delta z = \hat{z} - z$.

This experiment was repeated for 1 000 000 pseudorandom object points. The results are shown in Fig. 2 as a histogram of the range estimation error. The histogram has been scaled to unit area and superimposed with the theoretical probability density function, given by (10)–(12). The close match between the experimental histogram and the theoretical probability density function supports the validity of (10)–(12) as a representation of the range error distribution.

Using this same experiment, the average range error magnitude was computed. The experimental value of 42.71 m agrees well with the theoretical value of 42.32 m given by (14).

VII. Conclusion

A stochastic analysis of the quantization error in a stereo imaging system has been presented. The probability density function of the range estimation error and the expected value of the range error magnitude have been derived in terms of the various design parameters. In addition, the *relative range error* is proposed as a better measure of the range resolution than the percent range error when the depths in the scene lie within a narrow range. With these results, the designer of a stereo imaging system can more accurately determine how the choice of parameters will affect the expected range resolution.

Acknowledgment

Dr. J. D. Carothers deserves many thanks for her help in the proofreading of this paper.

References

[1] S. D. Blostein and T. S. Huang, "Error analysis in stereo determination of 3-D point positions," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, pp. 752–765, Nov. 1987.

[2] E. Krotkov and R. Kories, "Stereo ranging with verging cameras: A practical calibration procedure and error analysis," Dep. Comput. Inform. Sci., Univ. Pennsylvania, Philadelphia, Tech. Rep. MS-CIS-86-86, 1986.

[3] L. Matthies and S. A. Shafer, "Error modeling in stereo navigation," *IEEE J. Robotics Automat.*, vol. RA-3, pp. 239–248, June 1987.

[4] E. S. McVey and J. W. Lee, "Some accuracy and resolution aspects of computer vision distance measurements," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-4, pp. 646–649, Nov. 1982.

[5] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. New York: McGraw-Hill, 1984.

[6] C. C. Slama, Ed., *Manual of Photogrammetry*. Falls Church, VA: Amer. Soc. Photogrammetry, 1980.

[7] F. Solina, "Errors in stereo due to quantization," Dep. Comput. Inform. Sci., Univ. Pennsylvania, Philadelphia, Tech. Rep. MS-CIS-85-34, 1985.

[8] A. Verri and V. Torre, "Absolute depth estimate in stereopsis," *J. Opt. Soc. Amer.*, vol. 3, pp. 297–299, Mar. 1986.

Partial Shape Recognition: A Landmark-Based Approach

NIRWAN ANSARI AND EDWARD J. DELP

***Abstract*—When objects are occluded, many shape recognition methods that use global information will fail. To recognize partially occluded objects, we represent each object by a set of "landmarks." The landmarks of an object are points of interest relative to the object that have important shape attributes. Given a scene consisting of partially occluded objects, a model object in the scene is hypothesized by matching the landmarks of the model with those in the scene. A measure of similarity between two landmarks, one from the model and the other from the scene, is needed to perform this matching. In this correspondence we introduce a new local shape measure, sphericity. It will be shown that any invariant function under a similarity transformation is a function of the sphericity.**

To match landmarks between the model and the scene, a table of compatibility, where each entry in the table is the sphericity value derived from the mapping of a set of three model landmarks to a set of three scene landmarks, is constructed. A technique, known as hopping dynamic programming, is described to guide the landmark matching through the compatibility table. The location of the model in the scene is estimated with a least squares fit among the matched landmarks. A

Manuscript received July 22, 1988; revised November 14, 1989. Recommended for acceptance by W. E. L. Grimson.

N. Ansari was with the Computer Vision and Image Processing Laboratory, School of Electrical Engineering, Purdue University, West Lafayette, IN 47907. He is now with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102.

E. J. Delp is with the Computer Vision and Image Processing Laboratory, School of Electrical Engineering, Purdue University, West Lafayette, IN 47907.

IEEE Log Number 8933761.

0162-8828/90/0500-0470$01.00 © 1990 IEEE

heuristic measure is then computed to decide if the model is in the scene.

***Index Terms*—Affine transformation, dynamic programming, landmarks, occlusion, partial shape recognition.**

I. Introduction

Shape recognition is an important task in pattern recognition and computer vision. We will use the term *shape* to refer to the invariant geometrical properties of the relative distances among a set of static spatial features of an object. These static spatial features are known as the *shape features* of the object. A shape feature is classified as either a global or a local representation. A *global shape feature* represents the entire object region, such as the silhouette or contour of the object; *local shape features* represent portions of the object region, such as line segments, edges, and corners of the objects.

After extracting the shape features from a model and a scene, some sort of similarity or dissimilarity measure must be used to quantify the difference between the shape features. The similarity or dissimilarity measure is referred to as a *shape measure*. The shape measure should remain the same when the object is viewed at a different scale or at a different orientation. This does not suggest that size and orientation are not important for the shape recognition task. They are in fact important attributes that will be estimated either as a part of the shape recognition system, or as a separate task. Shape measures should thus be invariant to translation, rotation, and scaling.

A shape measure is classified as either a global or a local shape measure. A *global shape measure* quantifies the similarity between two entire objects; a *local shape measure* quantifies the similarity between portions of the objects. A global shape measure is derived from the global shape features of the objects; a local shape measure is derived from the local shape features. In this correspondence, we introduce a new local shape measure, sphericity. It will be shown that any invariant function under a similarity transformation must be a function of the sphericity. The problem we address is that of recognizing and locating planar objects that may be occluded or touching each other.

For the purpose of recognition, much of the visual data perceived by a human being is highly redundant. It has been suggested from the viewpoint of the human visual system [1] that some dominant points along an object contour are rich in information content and are sufficient to characterize the shape of the object. This concept of dominant points has been applied in the field of morphometrics [2] to study and observe the growth of biological objects. These dominant points of an object are usually referred to as the landmarks of the object. However, we will define the *landmarks* of an object as the points of interest of the object that have important shape attributes. Examples of landmarks are corners, holes, protrusions, and high curvature points. They can be problem specific based on *a priori* knowledge. For example, in medical imaging, landmarks could be the location of important bone joints. *Landmark-based shape recognition* is motivated by the above concept of dominant points. It uses landmarks as shape features to recognize objects in a scene. One of the merits of landmark-based shape recognition is that the extraction of the entire object contour is *not* required to achieve recognition. It only requires a landmark extractor that can detect and order the landmarks in a sequence that corresponds to consecutive points along the object boundary.

A landmark-based shape recognition system is shown in Fig. 1. Landmarks extracted from a model object and from the scene are referred to as *model landmarks* and *scene landmarks*, respectively. Properties of the landmarks of each model can be used to guide the extraction of the landmarks in the scene. The hypothesis of a model object in the scene is made by matching the model landmarks with the scene landmarks. Based on a decision strategy on how well the landmarks of each model are matched with those in the scene, objects in the scene are identified.

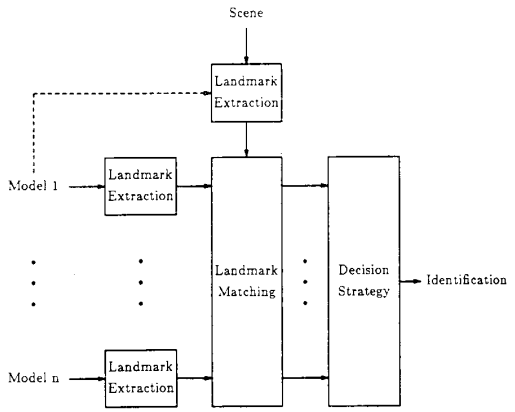


Fig. 1. A landmark-based shape recognition system.

Following a brief review of partial shape recognition methods in Section II, we will present an approach to extracting landmarks from an object contour in Section III. We will discuss the tasks of landmark matching, location estimation, and matching verification in Section IV. Experimental results will be presented in Section V.

II. Related Work

Recent work on 2-D partial shape recognition have exhibited an increasing interest in developing methods capable of recognizing objects when global information about the objects is not available. The recognition task can modeled as searching for a match between model and scene features. Commonly used features are holes and points [3]–[5], line segments [6]–[10], curve segments [11]–[15] or a combination of these features [16], [17]. The features are obtained by a preprocessing step such as edge detection, polygonal approximation, and corner extraction. We have taken a similar view by posing the recognition task as a landmark matching problem. Detailed literature reviews can be found in many of the above mentioned publications and in [18]. To provide a general overview of how the partial shape recognition task has been addressed, we will discuss the methods reported in [7], [8], [16], [17], [19]. We do not consider the problem of 3-D object recognition such as recognizing polyhedral objects [20]. A rather detailed survey of 3-D object recognition can be found in [21].

Bhanu and Faugeras [7] cast the shape matching problem as a segment matching problem. An object contour is first approximated by a polygon from which features such as the length of a segment, the slope of a segment, the angle between two adjacent segments, and the intervertice distance are computed. The sum of the weighted absolute differences of the feature values between a model and a scene segment is the shape measure between the two segments. This measure indicates the goodness of match between the two segments. A stochastic labeling scheme is then used to label each model segment either as one of the scene segments or NIL (no match).

Since this method uses relaxation labeling, it is computationally intensive. A good estimate of the initial assignment of the labels is important relative to the convergence of the algorithm and the validity of the result.

A simple technique to solve the occlusion problem has been proposed by Price [8]. The shape features of an object are the line segments of the approximated polygon of the object. Each model segment is then compared to every scene segment in terms of their

lengths, and the included angles between successive segments. If the length and the angle are within certain thresholds, the model segment is said to be compatible with the scene segment, and their orientation difference is stored in an array known as a disparity array. Since segments of an object are arranged sequentially along the object contour, segments between the model and the scene are likely to be matched in a sequence. The longest consecutive sequence of matched segments between the model and the scene corresponds to the longest compatible consecutive diagonal entries of the disparity array that have similar orientation differences. A transformation that aligns the model segments with the matched scene segments is evaluated. Applying this transformation to the model segments, disparity values based on the segment positions and orientations are updated and stored in the disparity array. The final matches between the model and the scene segments are determined by finding the longest compatible consecutive diagonal entries of the new disparity array.

Price's procedure is simple, but not computationally efficient since every entry of the disparity array has to be tested for the starting location of the longest sequence. Furthermore, the technique is sensitive to scale variation because the feature value, such as the length of a line segment, used in this technique is inherently scale dependent.

Bhanu and Ming [10] improve Price's approach by using the same disparity array but with a different matching process. The matching process first applies the *K-mean* clustering algorithm iteratively on the disparity array until the optimal number of clusters is found. It then checks for the elements of each cluster that are in sequential order, and finds the sequences. Several heuristics are included to determine the sequences. The process then clusters the sequence averages using the same clustering algorithm described above. The cluster which contains the largest number of sequences determines the final matches between the model and the scene segments. A confidence value which is the ratio of the cumulative length of the segments in the final match to the total length of all segments of the model is evaluated to verify the final match. Although it is computationally more efficient than Price's approach, it remains computationally expensive because of the iterative nature of the algorithm.

Gorman and Mitchell [19] represent an object contour by breaking the contour into contour segments. The breakpoints of the contour are the vertices which result from a polygonal approximation of the contour. Each contour segment is a portion of the object contour and consists of three consecutive vertices. It begins from a vertex which is considered as the first vertex and then ends at the third vertex along the object contour. The feature values of each contour segment are the Fourier coefficients derived from tracing along the segment from the beginning to the end and then back to the beginning of the segment. The shape measure between a model and a scene contour segment is the norm squared distance between the Fourier coefficients of the two segments. An intersegment distance table measuring the norm squared distances between the model and the scene contour segments is constructed. The table is augmented by repeating the rows. A backward dynamic programming procedure is then used to determine the minimum distance path starting from the first column to the last column of the augmented table. An entry along the minimum distance path that results from a diagonal transition corresponds to a match between the model and the scene segment, indicated by the row and the column index of the entry. This approach is not sensitive to scale variation because the Fourier coefficients have been normalized.

Lamdan *et al.* [16], [17] propose an approach to solve the occlusion problem by affine invariant matching. The concept can be applied to matching points, lines and curves. We will only discuss the point matching algorithm from which line and curve matching are based.

For each noncollinear triplet of model points, the coordinates of all other model points are computed using this triplet as an affine basis. Each such coordinate is used as an entry to a table (hash-table); each entry of the table contains the model and the triplet basis from which the coordinate is derived. Given a scene represented by a set of points, the affine coordinates of the scene points are similarly computed using an arbitrary triplet of scene points as a basis. Based on a voting procedure, each scene coordinate is then assigned a pair (*model, basis triplet*) to which it corresponds. If a certain pair scores a large number of votes, this pair is considered to correspond to the one chosen in the scene. If not, the process continues by checking another basis triplet. The transformation between the model and the scene can either be assumed to be the affine transformation between the chosen model and scene triplets [16] or be determined by a least squares affine transformation between the model and the scene points that match [17]. The complexity of the algorithm depends on the number of model points in the scene. It is bounded from above by $O(n^4)$ and from below by $O(n)$, where n is the number of scene points.

The shape recognition approach presented in this paper is different from the above methods in that the shape features of an object are the landmarks associated with the object. Instead of evaluating many feature values in order to characterize the similarity between two features, we use a function known as the sphericity to discriminate the dissimilarity between two landmarks. The sphericity along with its properties will be discussed in Section IV. In contrast to some partial shape recognition methods [3], [4], [8], [11], [12], our landmark matching algorithm is not sensitive to scale variation. The feature matching algorithm proposed is not iterative, and is based on what we call hopping dynamic programming to perform landmark matching.

III. Landmark Extraction

It is important to note that the entire contour of an object is not needed to use landmarks to achieve recognition. The approach only requires knowledge of the positions of the landmarks of the object in the image. It is necessary to impose a consistent ordering of the landmarks. If interior points of the object are used as landmarks, it is necessary to arrange them in a predefined order reflecting the shape and geometry of the object.

The main problem addressed in this paper is landmark matching for object recognition and *not* landmark extraction. To discuss how our method works it is necessary to first obtain the landmarks. In this section we propose a simple landmark extraction algorithm based on curvature. It is important to emphasize that landmark extraction should be considered as a preprocessing step and *other* landmarks that have a consistent ordering will work with our proposed method. The method is not tied directly to curvature points.

Points with high curvature along the object contour are features that can be used as landmarks. The contour, as in the case of a model, usually represents one object. However, in a general scene, when occlusion is allowed, the contour could represent merged boundaries of several objects. For illustrative purposes, in this correspondence we will only consider landmarks as points of high curvature along an object contour. Note that erroneous landmarks of objects in a scene may occur due to occlusion or noise in the scene. We will also assume that the images are obtained by orthographic projection, and the silhouette of an object region is either given or can be acquired from a segmentation algorithm.

Due to the discrete boundary representation and quantization error, false local concavities and convexities along a boundary are introduced. Smoothing is thus necessary to reduce false concavities and convexities. A Gaussian filter, which has been shown to be an ideal smoothing filter for numerical differentiation [22], is used.

A planar curve can be represented by a set of points in parametric form, $(x(t), y(t)) \in \mathbb{R}^2$, where t is the path length along the curve. Smoothing the curve with a Gaussian filter is thus convolving $x(t)$ and $y(t)$, respectively, with a one-dimensional Gaussian filter,

$$\eta(t, \omega) = \frac{1}{\sqrt{2\pi}\omega} e^{(1/2)(t/\omega)^2},$$

where ω is the width of the filter. As we "traverse" along a curve in increasing values of the path length, a positive curvature corresponds to a concavity on our left, and a negative curvature corresponds to a concavity on our right. The extreme (positive maximum and negative minimum) curvature points of a boundary which has been smoothed by a Gaussian filter with a large ω are stable with respect to orientation and scaling; i.e., their corresponding locations along the unsmoothed boundary remains relatively unchanged when the boundary is rotated, or scaled within a reasonable range. We will refer to these stable local extreme curvature points as the *cardinal curvture points*.

Unlike Asada and Brady's curvature primal sketch [23], we do not match the locations of the extrema from different scales to determine the dominant points. Our procedure for detecting cardinal curvature points is straightforward. Given a library of object contours, each contour is first smoothed by a Gaussian filter with various widths (scales). Note that the number of extreme curvature points of an object can vary as the range of scales (widths) varies. The cardinal curvature points of the object are the extreme curvature points of the object that remains unchanged for a *range* of ω; at the same time, this range of ω shares a common range of ω that is used to obtain the cardinal curvature points of other objects in the library. Points along the original unsmoothed boundary that correspond to the cardinal curvature points are considered as the landmarks of the object. The range of ω of each object is determined off-line. For example, as shown in Table I, the cardinal curvature points of the wire stripper shown in Fig. 2(a) can be obtained by smoothing the contour with any ω that is in the range of 16.5-36.5. Landmarks obtained for other objects by this approach are shown in Fig. 2, with their corresponding range of ω in Table I.

TABLE I
RANGE OF ω USED TO OBTAIN THE LANDMARKS OF A LIBRARY OF OBJECTS BASED ON CARDINAL CURVATURE POINTS

Models	Figure	Range
wire stripper	2a	16.5-36.5
wrench	2b	14.5-24
specialty plier	2c	14-40.5
needle-nose plier	2d	14.5-(>100)
wire cutter	2e	16-28.5
spacecraft	2f	19.5-25.5
Borneo	2g	19.5-24
Halmahera	2h	20-23
Luzon	2i	14-19.5
Mindanao	2j	17.5-21.5
New Guinea	2k	18-21
Sulawesi	2l	19.5-29

IV. SPHERICITY

The sphericity of a triangular transformation which maps a triangle to another triangle is a measure of the similarity between the two triangles. Under the triangular transformation, the inscribed circle of one triangle is mapped into an inscribed ellipse of the other triangle. As shown in Fig. 3, the sphericity is defined as the ratio of the geometric mean to the arithmetic mean of the lengths of the principal axes of the inscribed ellipse; i.e., the sphericity $= (2\sqrt{d_1 + d_2})/(d_1 + d_2)$. If the two triangles are similar, the sphericity is one. The less similar the two triangles are, the smaller is the value of the sphericity. If the vertices of one triangle are considered as the coordinates of three consecutive landmarks belonging to a model, and the vertices of the other triangle as those belonging to a scene, the sphericity is thus a measure of the similarity between the set of three model landmarks and the set of three scene landmarks. Among various properties of sphericity that we will discuss, we will show that any invariant function under a similarity transformation is a function of the sphericity. The triangular transformation is uniquely defined by an affine transformation [24].

Definition 1: An affine transformation is the mapping of $\boldsymbol{x}$ to $\boldsymbol{u}$, where $\boldsymbol{x}, \boldsymbol{u} \in \mathbb{R}^2$, defined by:

$$\boldsymbol{u} = A\boldsymbol{x} + \boldsymbol{t}, \tag{1}$$

where

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix}, \boldsymbol{u} = \begin{bmatrix} u \\ v \end{bmatrix}, \boldsymbol{t} = \begin{bmatrix} e \\ f \end{bmatrix}, A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \text{ and } \det(A) \neq 0.$$

Coefficients of the affine transformation which maps one triangle into another, as shown in Fig. 3, are computed using the following equations:

$$\begin{bmatrix} a \\ b \\ e \end{bmatrix} = B^{-1} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \text{ and } \begin{bmatrix} c \\ d \\ f \end{bmatrix} = B^{-1} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}, \tag{2}$$

where

$$B = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix}.$$

(u_i, v_i) are the image points of the points (x_i, y_i), $i = 1, 2$, and 3, under the transformation described by (1). Since vertices of a triangle are noncollinear, $\det(B) \neq 0$ and B^{-1} exists. Assuming we know the vertices of the first triangle, we can compute the parameters of the inscribed circle (its radius and center). The inscribed circle is then mapped through the transformation into an ellipse, and the sphericity can then be computed.

A. Mathematical Definition and Properties of the Sphericity

We have presented an intuitive description of the sphericity of a triangular transformation. We will generalize its definition to a diffeomorphism.[1] Desirable properties of the sphericity will also be derived.

Definition 2: The sphericity of a diffeomorphism, $g: \Omega \to \bar{\Omega}$, ($\Omega$, $\bar{\Omega} \subset \mathbb{R}^n$), for $x \in \Omega$, is defined as

$$Y_g(x) = \frac{\left(\det\left(g''(x)\, g'(x)\right)\right)^{1/n}}{\left(\frac{1}{n}\operatorname{tr}\left(g''(x)\, g'(x)\right)\right)} \tag{3}$$

where

$g'(x)$ is the derivative of $g(x)$,
$g''(x)$ is the transpose of $g'(x)$, and
det ( ) and tr ( ) are the determinant and the trace of a matrix, respectively.

For notational convenience, we simply denote $Y_g(x)$, $g'(x)$, and $g''(x)$ as Y_g, g', and g'', respectively. Note that g' maps a unit ball in Ω into an ellipsoid in $\bar{\Omega}$. Note also that $g''g'$ is positive definite.

Lemma 1: The sphericity can be expressed in terms of the eigenvalues of $g''g'$:

$$Y_g = \frac{(\lambda_1\lambda_2 \cdots \lambda_n)^{1/n}}{\left(\frac{1}{n}(\lambda_1 + \lambda_2 + \cdots + \lambda_n)\right)} \tag{4}$$

where λ_i, $i = 1, 2, \cdots, n$, are the eigenvalues of $g''g'$.

Proof: the result is obtained by noting the following:

$$\begin{aligned} \det(g''g') &= \det(\Phi' D \Phi) \text{ (spectral decomposition)} \\ &= \det(D\Phi'\Phi) = \det(D) = \lambda_1\lambda_2 \cdots \lambda_n, \end{aligned}$$

and

$$\begin{aligned} \operatorname{tr}(g''g') &= \operatorname{tr}(\Phi' D \Phi) = \operatorname{tr}(D\Phi'\Phi) = \operatorname{tr}(D) \\ &= \lambda_1 + \lambda_2 + \cdots + \lambda_n. \end{aligned}$$

□

[1] A *diffeomorphism* is a continuous one-to-one mapping such that the inverse mapping is also continuous, and both the mapping and its inverse have continuous partial derivatives.

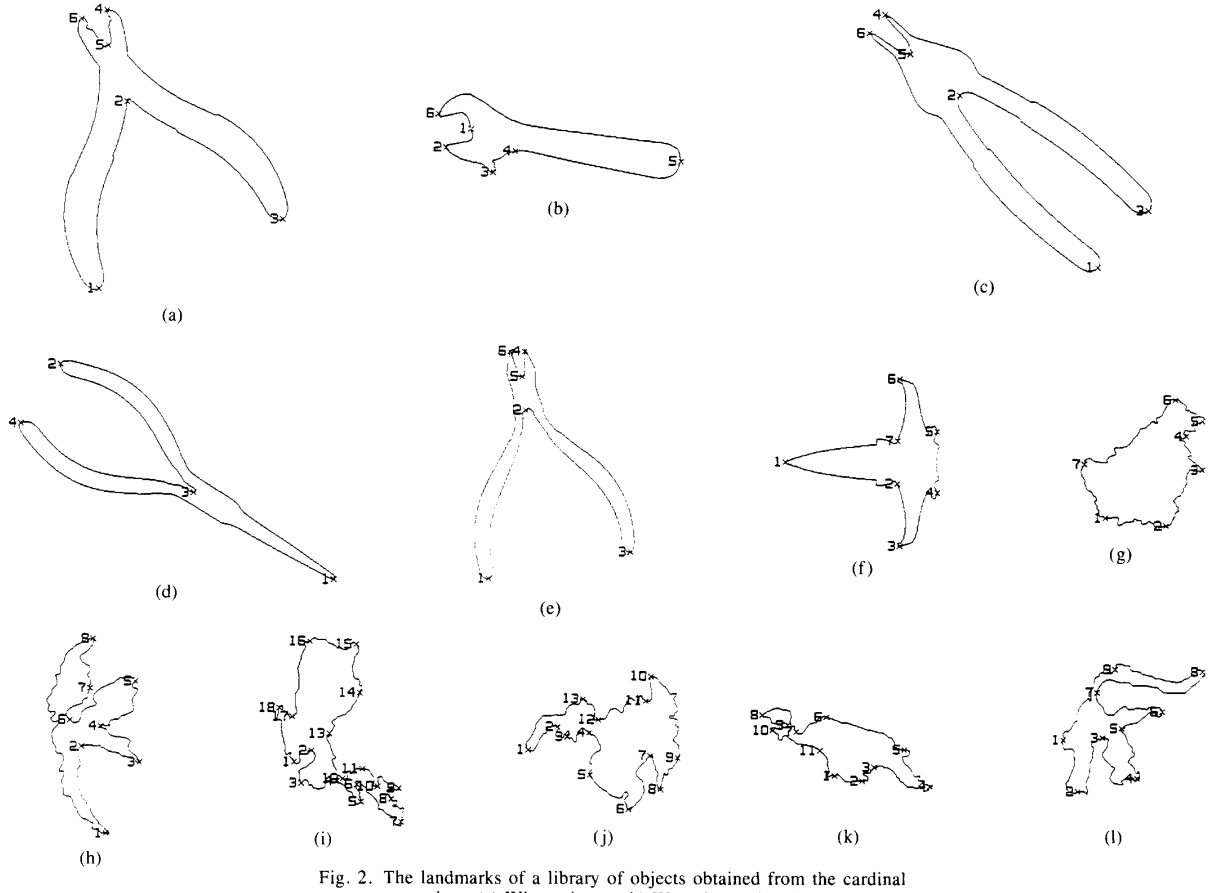


Fig. 2. The landmarks of a library of objects obtained from the cardinal curvature points. (a) Wire stripper. (b) Wrench. (c) Specialty plier. (d) Needle-nose plier. (e) Wire cutter. (f) Spacecraft. (g) Island of Borneo. (h) Island of Halmahera. (i) Island of Luzon. (j) Island of Mindanao. (k) Island of New Guinea. (l) Island of Sulawesi.

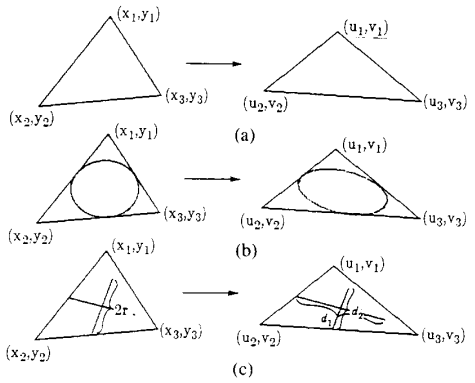


Fig. 3. Representation of a mapping from a triangle to another triangle. (a) Original triangles. (b) Mapping from the inscribed circle to an inscribed ellipse. (c) Mapping of the principal axes. Sphericity = $2\sqrt{d_1 + d_2}/d_1 + d_2$.

Note that the eigenvalues of $g''g'$ correspond to the lengths of the semiaxes of the transformed ellipsoid. Since the geometric mean of a sequence of positive real numbers is always greater than or equal to the arithmetic mean of the same sequence of numbers [25], the sphericity thus takes on values in [0, 1]. If $Y_g(x) = 1$ for all $x \in \Omega$, $g(x)$ is a conformal mapping.

Lemma 2: The sphericity of a diffeomorphism described in Definition 2 is invariant when the diffeomorphism undergoes a group of transformations, G = {translation, rotation, scaling}.

Proof: Let g be a diffeomorphism, and assume g is transformed by G with the result being $\bar{g}$, that is,

$$\bar{g} = \alpha Hg + h,$$

where

α is the scaling factor,
H is the rotation matrix, and
h is the translation vector.

Thus,

$$\bar{g}''\bar{g}' = (\alpha Hg')^t(\alpha Hg') = \alpha^2 g''(H'H)g' = \alpha^2 g''g'.$$

Using Definition 2, $Y_{\bar{g}} = Y_g$. □

Definition 3: Denote G as a group of transformations from a space μ into itself. A function $\phi(u)$ on μ is said to be a maximal invariant under G [26] if it is invariant under G, and

if $\phi(u_1) = \phi(u_2)$, there exists $\zeta \in G$ such that $u_1 = \zeta u_2$,

where $u_1 = \zeta u_2$ indicates that ζ maps u_1 to u_2.

Using the above definition, we can prove the following theorem.

Theorem 1: Let $\boldsymbol{u}$ be a set of three two-dimensional points, that is, $\boldsymbol{u} \in \mu = \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2$.

Denote $\boldsymbol{Y}(\boldsymbol{u})$ as the sphericity derived from the affine transformation of a set of three fixed points, $\boldsymbol{x}$, to $\boldsymbol{u}$. $\boldsymbol{Y}(\boldsymbol{u})$ is a maximal invariant under $\boldsymbol{G}$ = {translation, rotation, scaling}.

Proof: Note that we have used $\boldsymbol{Y}(\boldsymbol{u})$ to indicate that $\boldsymbol{u}$ is transformed by $\boldsymbol{G}$. The sphericity of a diffeomorphism is invariant under $\boldsymbol{G}$ from Lemma 2. Since an affine transformation is a diffeomorphism, the sphericity of a triangular transformation is invariant under $\boldsymbol{G}$. To show that the sphericity of a triangular transformation is a maximal invariant under $\boldsymbol{G}$, it is sufficient to show that if $\boldsymbol{u}_1$ and $\boldsymbol{u}_2 \in \mu$, then

$$\boldsymbol{Y}(\boldsymbol{u}_1) = \boldsymbol{Y}(\boldsymbol{u}_2) \text{ implies } \boldsymbol{u}_2 = \zeta \boldsymbol{u}_1 \qquad \text{for some } \zeta \in \boldsymbol{G}.$$

Given $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ in μ, there exists two affine transformations that map the set of three fixed points $\boldsymbol{x}$ to $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$, respectively. Let A_1 and t_1 be the linear part and the translation part of the affine transformation that maps $\boldsymbol{x}$ to $\boldsymbol{u}_1$, and likewise for A_2 and t_2. From Lemma 1, if $\boldsymbol{Y}(\boldsymbol{u}_1) = \boldsymbol{Y}(\boldsymbol{u}_2)$, then the eigenvalues of $A_1^t A_1$ are multiples of the eigenvalues of $A_2^t A_2$. That is,

$$A_2 = k\Phi A_1$$

for some constant k, and Φ is an orthogonal matrix. Note that the column vectors of Φ form an orthogonal basis in $\mathbb{R}^2$. Φ corresponds to a rotation matrix and k corresponds to the scaling. Finally, the translation part does not contribute to the evaluation of the sphericity.

Therefore,

$$\boldsymbol{u}_2 = \zeta \boldsymbol{u}_1. \qquad \square$$

The importance of a maximal invariant is shown by the following theorem.

Theorem 2: If $\phi(\boldsymbol{u})$ on μ is a maximal invariant under $\boldsymbol{G}$, then $\psi(\boldsymbol{u})$ is invariant under $\boldsymbol{G}$ if and only if ψ is a function of $\phi(\boldsymbol{u})$.

Proof: The proof can be found in [26]. $\square$

From Theorem 2, we can conclude that any invariant function under $\boldsymbol{G}$ derived from a triangular transformation is a function of the sphericity.

Lemma 3: The sphericity of an affine transformation defined by (1), for $|A| > 0$, is:

$$Y_g = \frac{t_1^2 + t_3^2 - (t_2^2 + t_4^2)}{t_1^2 + t_2^2 + t_3^2 + t_4^2}, \tag{5}$$

where

$$t_1 = a + d, \quad t_2 = a - d,$$

$$t_3 = b - c, \quad \text{and} \quad t_4 = b + c.$$

Note that (5) expresses the sphericity Y_g in terms of the coefficients of the affine transformation.

Proof: From (1) and (3),

$$Y_g = \left[\frac{\det(A^t A)}{\left(\frac{1}{2}\operatorname{tr}(A^t A)\right)^2} \right]^{1/2}$$

$$= \left[\frac{\det\left(\begin{bmatrix} a^2 + c^2 & ab + cd \\ ab + cd & b^2 + d^2 \end{bmatrix} \right)}{\left(\frac{1}{2}(a^2 + b^2 + c^2 + d^2)\right)^2} \right]^{1/2}$$

$$= \frac{2(ad - bc)}{(a^2 + b^2 + c^2 + d^2)} = \frac{t_1^2 + t_3^2 - (t_2^2 + t_4^2)}{t_1^2 + t_2^2 + t_3^2 + t_4^2}. \qquad \square$$

Lemma 3 provides a simple equation for computing the sphericity of a triangular transformation, and will be used in the next section to determine the sensitivity of the sphericity with respect to distortion. Other properties of the sphericity are discussed in [18].

B. The Probability Density Function of the Sphericity

Let $\{(u_1, v_1), (u_2, v_2), (u_3, v_3)\}$ be the coordinates of a sequence of three consecutive landmarks belonging to a scene, and $\{(x_1, y_1), (x_2, y_2), (x_3, y_3)\}$ be three consecutive landmarks belonging to a model. The sphericity of the triangular transformation which maps $\{(x_1, y_1), (x_2, y_2), (x_3, y_3)\}$ to $\{(u_1, v_1), (u_2, v_2), (u_3, v_3)\}$ determines how well the model landmarks match the scene landmarks. If the object in the scene is the rotated, translated, scaled, or a distorted version of the model, how well do the associated landmarks match each other? In other words, is the shape measure (sphericity) robust with respect to rotation, translation, scaling, and distortion? It has been shown in Section IV-A that the sphericity is rotation, translation, and scale invariant. In this section, we will discuss the relative robustness of the sphericity with respect to distortion in the scene landmark locations. If the set of three scene landmarks are duplicates of the three model landmarks, the sphericity derived from the mapping of the model landmarks to the scene landmarks should be one. We will assume that the distortion in the scene landmarks can be modeled as "noise" added to the locations of the model landmarks by the following:

$$u_i = x_i + n_i, \qquad i = 1, 2, 3,$$

and

$$v_i = y_i + n_{i+3}, \qquad i = 1, 2, 3,$$

where n_i, $i = 1, 2, \cdots, 6$, are independent identically distributed (i.i.d.) normal (Gaussian) random variables with mean zero and standard deviation σ.

That is,

$$u_1 \sim \boldsymbol{n}(x_1, \sigma^2), \quad v_1 \sim \boldsymbol{n}(y_1, \sigma^2)$$

$$u_2 \sim \boldsymbol{n}(x_2, \sigma^2), \quad v_2 \sim \boldsymbol{n}(y_2, \sigma^2),$$

and

$$u_3 \sim \boldsymbol{n}(x_3, \sigma^2), \quad v_3 \sim \boldsymbol{n}(y_3, \sigma^2),$$

where $\boldsymbol{n}(\mu, \sigma^2)$ denotes a normal probability density function with mean μ and standard deviation σ.

The i.i.d. normal random variables are used to make the analysis tractable. With the above assumptions, the sphericity is a random variable. We want to determine the probability density function, the mean, and the variance of the sphericity. We would hope that the sphericity has a mean close to 1 and variance close to 0 when σ is small. This leads to the following theorem.

Theorem 3: If the set of points (x_1, y_1), (x_2, y_2), (x_3, y_3) form an equilateral triangle, the sphericity based on the above model of distortion has the following probability density function, mean, and variance:

$$f_{Y_g}(v_g) = \frac{1}{2}\beta\left(\frac{v_g - 1}{2}; 1, 1, \rho\right) \tag{6}$$

$$E(\boldsymbol{Y}_g) = 1 - \frac{2}{\rho} + \frac{2}{\rho^2} - \frac{2}{\rho^2}e^{-\rho} \tag{7}$$

$$\operatorname{var}(\boldsymbol{Y}_g) = \frac{4}{\rho^2}\left(1 - \frac{2}{\rho} - \frac{1}{\rho^2} + 2e^{-\rho}\left(1 + \frac{1}{\rho} + \frac{1}{\rho^2} + \frac{e^{-\rho}}{\rho^2}\right)\right) \tag{8}$$

where

$f_{Y_g}(\cdot)$ is the probability density function of the sphericity $\boldsymbol{Y}_g$,
v_g is the independent variable of the function $f_{Y_g}(\cdot)$, and
$\beta(\cdot; 1, 1, \rho)$ denotes the noncentral Beta probability density function [27] with 2, 2 degrees of freedom, and noncentrality ρ, with $\rho = (\text{sidelength})^2/2\sigma^2$, where "sidelength" refers to the length of a side of the equilateral triangle.

Proof: A detailed proof can be found in [18].

From (7) and (8), we see that if $\sigma^2 \to 0$, then,

$$E(\boldsymbol{Y}_g) \to 1 \quad \text{and} \quad \operatorname{var}(\boldsymbol{Y}_g) \to 0.$$

Also note that

$$\text{if } \frac{1}{\rho} < 0.02 \left(\frac{\sigma}{\text{sidelength}} < 0.1 \right),$$
$$\text{then var } (Y_g) < 0.015 \text{ and } E(Y_g) > 0.96.$$

This means that if the standard deviation of the i.i.d. normal random variables, the distortion, is less than 10% of the sidelength of the equilateral triangle, the mean of the sphericity is greater than 0.96 and the variance is less than 0.015. Since the sphericity of a triangular transformation indicates the similarity between the two triangles formed from the model and the scene landmarks, respectively, the two triangles are less similar as the distortion increases. We thus expect that, as the distortion increases (σ increases), the mean of the sphericity decreases from 1, and the variance of the sphericity increases, as indicated by the above equations.

We have shown the probability density function of the sphericity in closed form for the special case of the equilateral triangle. We empirically estimate the probability density function, the mean, and the standard deviation of the sphericity for several other types of triangles using histograms. Each type is specified by an angle with a fixed height and a fixed base length such that an angle of 60° corresponds to an equilateral triangle, as shown in Fig. 4. We refer to the smallest perpendicular distance of a triangle as the smallest perpendicular distance from a vertex to the opposite side of the triangle. Each of the i.i.d. normal random variables that is used for modeling the distortion is assumed to have zero mean, and standard deviation equal to a percentage of the smallest perpendicular distance of the triangle. Ten thousand samples are used for each estimate.

The sphericity is distributed on [0, 1], which is quantized into 50 regions for the cases studied. Figs. 5 and 6 show the estimated probability density function of the sphericity for two types of triangles specified by their angles. Each value of the "noise level" corresponds to a percentage of the smallest perpendicular distance of the triangle; the length corresponding to this percentage is used as the standard deviation of the zero mean i.i.d. normal random variables used to model the distortion. The estimated mean of the sphericity for various types of triangles and "noise levels" is shown in Fig. 7, and the estimated standard deviation is shown in Fig. 8. With a small noise level, the sphericity has an estimated mean close to 1, and standard deviation close to 0. The plots of the estimated probability density function of the sphericity for other types of triangles are similar to those shown here.

We conclude from the above analysis that the sphericity is relatively robust with respect to distortion in the sense that a small perturbation in the landmarks does not significantly affect the value of the sphericity.

V. The Use of the Sphericity

The hypothesis of a model object in a scene is made by matching the model landmarks with the scene landmarks. The landmark matching task is performed by a procedure known as hopping dynamic programming. The location of the object in the scene is then estimated by a least squares fit. A heuristic measure based on the least squared error of the fit is used to verify the hypothesis.

A. Landmark Matching by Hopping Dynamic Programming

Let $\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$ be the coordinates of a sequence of landmarks associated with a model, and $\{(u_1, v_1), (u_2, v_2), \cdots, (u_m, v_m)\}$ be those associated with a scene. Note that n is the number of model landmarks, and m is the number of scene landmarks. The goodness of match between the ith model landmark and the jth scene landmark is given by the sphericity (5) derived from the triangular transformation mapping $\{(x_{i-1}, y_{i-1}), (x_i, y_i), (x_{i+1}, y_{i+1})\}$ to $\{(u_{j-1}, v_{j-1}), (u_j, v_j), (u_{j+1}, v_{j+1})\}$. A mapping is said to be *orientation or sense reversing* [28] if the Jacobian of the mapping is negative. It can be seen from (5) that the value of the sphericity is negative if the triangular transformation is sense reversing.

A table of compatibility is constructed between the sequence of model landmarks and the sequence of scene landmarks. The row

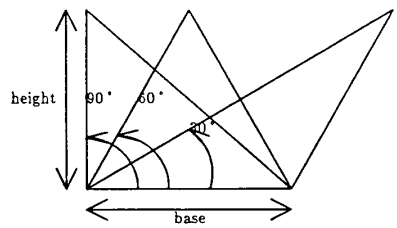


Fig. 4. Types of triangles used for estimating the probability density function of the sphericity, each specified by an angle.

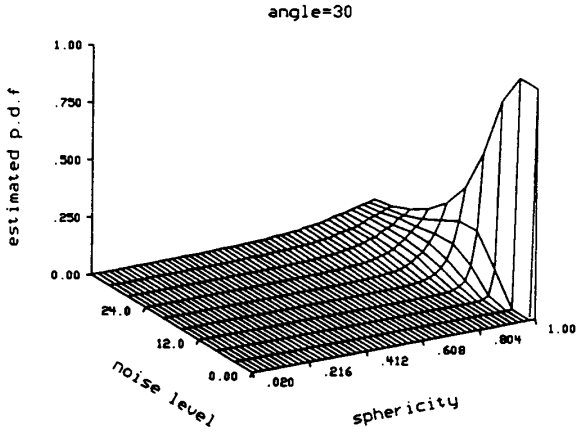


Fig. 5. Estimated probability density function of the sphericity for angle = 30° at various noise levels. Each value of the "noise level" corresponds to a percentage of the smallest perpendicular distance of the triangle. The length corresponding to the percentage value is used as the standard deviation of the zero mean i.i.d. normal random variables used for modeling the distortion in the landmarks.

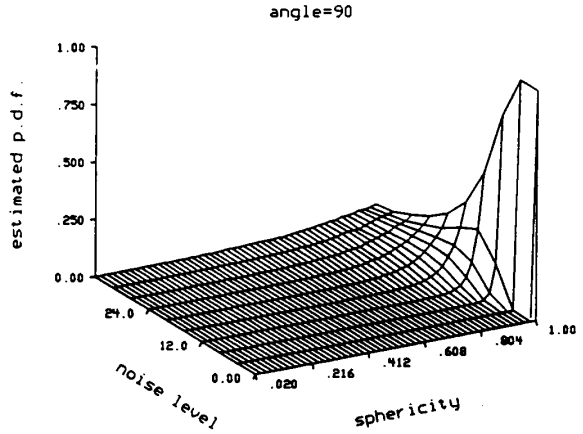


Fig. 6. Estimated probability density function of the sphericity for angle = 90° at various noise levels. Each value of the "noise level" corresponds to a percentage of the smallest perpendicular distance of the triangle. The length corresponding to the percentage value is used as the standard deviation of the zero mean i.i.d. normal random variables used for modeling the distortion in the landmarks.

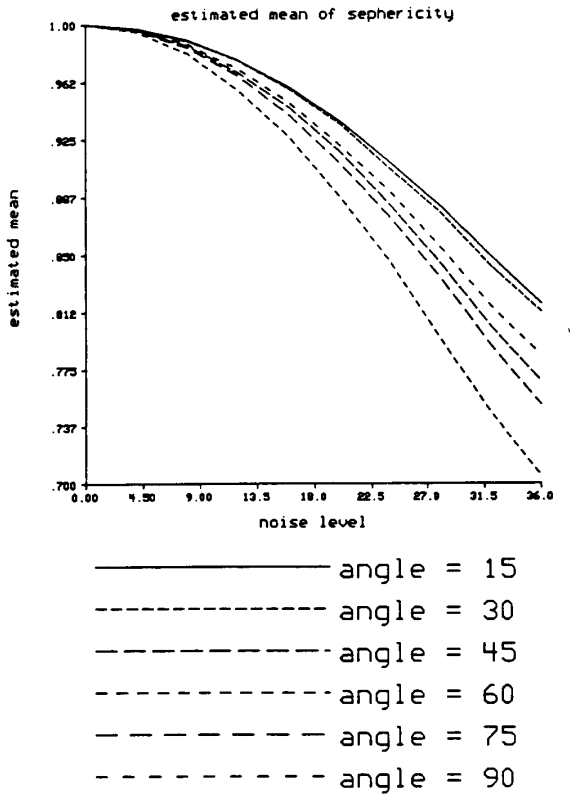


Fig. 7. Estimated mean of the sphericity corresponding to different types of triangles specified by the angles 15°, 30°, 45°, 60°, 75°, and 90°.

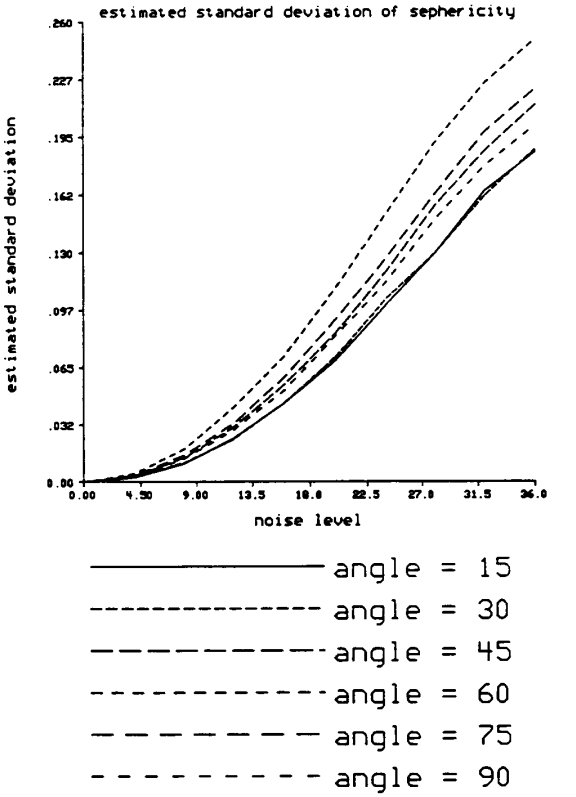


Fig. 8. Estimated standard deviation of the sphericity corresponding to different types of triangles specified by the angles 15°, 30°, 45°, 60°, 75°, and 90°.

index i corresponds to a model landmark while the column index j corresponds to a scene landmark. The (i, j) entry of the table is the sphericity value of the triangular transformation mapping the ith model landmark and its two adjacent landmarks to the jth scene landmark and its two adjacent landmarks. Consider a simple example of a scene where there are two objects overlapping each other as shown in Fig. 9. The extracted landmarks in the scene are based on the cardinal curvature points using $\omega = 20$. A table of compatibility between the wire stripper [Fig. 2(a)] and the scene (Fig. 9) is shown in Table II(a). Since the landmarks of an object are obtained by tracing sequentially along the object boundary, it is likely that matches between the model and scene landmarks correspond to a sequence of high-valued entries that are diagonal to each other in the table. This sequence will correspond to a path in the table. A brute-force approach to finding such a sequence is impractical. We will instead formulate a dynamic programming procedure to achieve the matching.

Our matching procedure is somewhat similar to the feature matching algorithm of [19]. Gorman and Mitchell [19] use a backward dynamic programming procedure to find a minimum distance path from the first column to the last column of their augmented intersegment distance table. Their assumption that the path must make use of all the scene features is inadequate because the scene may have extraneous or missing features due to occlusion. Instead of this assumption, we will only require that our path covers the range of either all the model landmarks or all the scene landmarks; i.e., the path traverses through either all the rows or all the columns of the table of compatibility. Neither the starting point nor the destination point of a path which corresponds to a sequence of matches between the scene and model landmarks is known. Instead, a *support entry*, which is an entry in the table that provides strong evi-

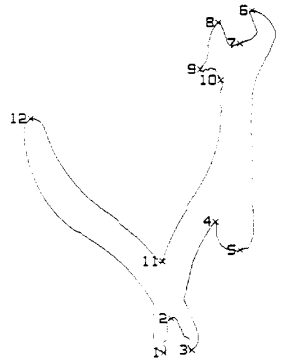


Fig. 9. A scene which consists of a wire stripper and a wrench overlapping each other. Each landmark is labeled and indicated by an "×."

dence of a true match between a model and a scene landmark, is used to guide the matching process. The evidence is strong if the entry as well as its diagonal neighboring entries have sphericity values close to one. That is, the model landmark and its neighboring landmarks match well locally with the scene landmark and its neighboring landmarks. Denote $s(i, j)$ as the sphericity value at the (i, j) entry of the table. The (i, j) entry of the table is said to be the support entry of the table if the sum $s(i-1, j-1) + s(i, j) + s(i+1, j+1)$ is maximum. In the example shown in Table II(a), the support entry can either be entry (3, 12) or (4, 1). Since the sphericity is a local similarity measure between a model and a

TABLE II
AN EXAMPLE OF PERFORMING LANDMARK MATCHING BY HDP. (a) THE TABLE OF COMPATIBILITY BETWEEN THE WIRE STRIPPER AND THE SCENE SHOWN IN FIG. 9. (b) THE RESULT OF PERFORMING HDP USING (3, 12) AS THE SUPPORT ENTRY. (c) THE MAXIMUM VALUE PATH IS INDICATED BY 1'S.

Model												
1	0.07	-0.37	0.08	-0.20	0.04	0.04	-0.31	0.19	0.17	-0.03	-0.31	0.16
2	-0.16	0.94	-0.40	0.41	-0.20	-0.08	0.98	-0.86	-0.62	0.20	1.00	-0.34
3	0.03	-0.44	0.21	-0.09	0.09	0.02	-0.28	0.32	0.12	-0.07	-0.34	1.00
4	1.00	-0.12	0.03	-0.58	0.02	0.72	-0.18	0.09	0.30	-0.02	-0.15	0.03
5	-0.12	1.00	-0.37	0.33	-0.17	-0.06	0.86	-0.76	-0.45	0.16	0.93	-0.48
6	0.02	-0.15	0.60	-0.04	0.99	0.01	-0.15	0.30	0.08	-0.54	-0.17	0.07
	1	2	3	4	5	6	7	8	9	10	11	12

Scene

(a)

Model						
1	2.00	2.00	1.00	0.00	0.00	0.00
2	2.00	2.00	1.00	0.00	0.00	0.00
3	1.00	1.00	1.00	1.00	1.00	1.00
4	0.00	0.00	1.00	1.99	1.99	1.99
5	0.00	0.00	1.00	1.99	2.99	2.99
6	0.00	0.00	1.00	1.99	2.99	3.60
	10	11	12	1	2	3

Scene

(b)

Model						
1	1	0	0	0	0	0
2	1	1	0	0	0	0
3	0	0	1	0	0	0
4	0	0	0	1	0	0
5	0	0	0	0	1	0
6	0	0	0	0	0	1
	10	11	12	1	2	3

Scene

(c)

scene landmark, the overall goodness of match between the model and the scene is determined by the sum of the sphericity values of those landmarks that match each other. The sequence of matches should correspond to a path in the table that passes through the support entry and maximizes the sum of the sphericity values of the path with the following two constraints:

1) a model landmark cannot match with more than one scene landmark, and

2) a scene landmark cannot match with more than one model landmark.

From the above two constraints, a vertical or a horizontal transition of the path should not be considered as a match between the model and the scene landmark.

Unlike the classical shortest path problem [29], we want to search for a path that passes through the support entry, rather than from a starting point to a destination point, or vice versa. Since backward dynamic programming is applicable when the destination point is available, and forward dynamic programming is applicable when the starting point is available [29], the support entry can be treated as both a starting and a destination point. That is, we work both forward and backward from the support entry.

Denote (k, l) as the support entry, $a_b(i, j)$ as the accumulated sum of the sphericity values from the (k, l) to the (i, j) entry in the backward procedure, and $a_f(i, j)$ as the accumulated sum of the sphericity values from the (k, l) to the (i, j) entry in the forward procedure.

Treating the support entry as the destination point, we have the following set of transition rules for the backward procedure:

1) $a_b(i-1, j-1) = \max\{a_b(i, j) + s(i-1, j-1), a_b(i-1, j), a_b(i, j-1)\}$
2) $a_b(i-1, l) = \max\{s(i, l), s(i-1, l)\}$
3) $a_b(k, j-1) = \max\{s(k, j), s(k, j-1)\}$
4) $a_b(k, l) = s(k, l)$.

A diagonal transition according to rule 1) implies a possible match between the $(i-1)$th model landmarks and the $(j-1)$th scene landmark, and hence the sphericity value at $(i-1, j-1)$ is added to the accumulated sum of the sphericity at (i, j) to produce the accumulated sum of the sphericity at $(i-1, j-1)$. Since a horizontal or a vertical transition does not constitute a match, the accumulated sum of the sphericity remains the same as before the transition. Rules 2) and 3) are the boundary conditions. Rule 4) is the initial condition. To account for the periodic nature of the landmarks to be matched, when $i - 1 < 1$, the value of $i - 1$ is replaced by $n + i - 1$; when $j - 1 < 1$, the value of $j - 1$ is replaced by $m + j - 1$.

Treating the support entry as the starting point, we have the following set of transition rules for the forward procedure:

1) $a_f(i+1, j+1) = \max\{a_f(i, j) + s(i+1, j+1), a_f(i+1, j), a_f(i, j+1)\}$
2) $a_f(i+1, l) = \max\{s(i, l), s(i+1, l)\}$
3) $a_f(k, j+1) = \max\{s(k, j), s(k, j+1)\}$
4) $a_f(k, l) = s(k, l)$.

Again, according to Rule 1), a diagonal transition implies a possible match between the $(i+1)$th model landmark and the $(j+1)$th scene landmark, and hence the accumulated sum of the sphericity at $(i+1, j+1)$ is obtained by summing the sphericity value at $(i+1, j+1)$ and the accumulated sum of the sphericity at (i, j). Similarly, rules 2) and 3) are the boundary conditions, and rule 4) is the initial condition. Again, to account for the periodic nature of the landmarks to be matched, when $i + 1 > n$, the value of $i + 1$ is replaced by $i + 1 - n$; when $j + 1 > m$, the value of $j + 1$ is replaced by $j + 1 - m$.

We now need to address how to switch between the forward and backward procedure. Taking a forward and a backward step alternately is not a good strategy because matches are not usually equally divided between the forward and the backward path. Let $(\bar{i}, \bar{j})$ be the entry which the backward procedure has reached at the present stage, and $(\hat{i}, \hat{j})$ be the entry which the forward procedure has reached at the present stage. We define the backward average sphericity value at entry $(\bar{i}, \bar{j})$ as $a_b(\bar{i}, \bar{j})$ divided by the number of transitions made by the backward procedure traversing from entry (k, l) to entry $(\bar{i}, \bar{j})$ of the table. Similarly, we define the forward average sphericity value at entry $(\hat{i}, \hat{j})$ as $a_f(\hat{i}, \hat{j})$ divided by the number of transitions made by the forward procedure traversing from entry (k, l) to entry $(\hat{i}, \hat{j})$ of the table. The procedure which has a larger average sphericity proceeds one stage. That is, if the backward average sphericity value at entry $(\bar{i}, \bar{j})$ is larger than the forward average sphericity value at entry $(\hat{i}, \hat{j})$, the backward procedure will proceed to entry $(\bar{i} - 1, \bar{j} - 1)$; otherwise, the forward procedure will proceed to $(\hat{i} + 1, \hat{j} + 1)$. In other words, the procedure having a more promising path of matches proceeds one stage. The algorithm continues in this fashion until the combined path of both the forward and the backward procedures covers the range of either all the model landmarks or all the scene landmarks. The combined path is known as the *maximum value path*. We call this procedure hopping dynamic programming (HDP).

Continuing from the earlier example, and using entry (3, 12) as the support entry, HDP yields the result shown in Table II(b). Each entry of the upper left portion of the table represents the accumulated sum of the sphericity at that entry resulting from the backward dynamic programming procedure. Likewise, each entry of the lower right portion of the table represents the accumulated sum of the sphericity at that entry resulting from the forward dynamic programming procedure. The resulting maximum value path is shown in Table II(c).

After determining the path, several heuristics are used to further refine the matches between the model and the scene landmarks along the path. From the two constraints mentioned earlier, entries along the path that result from horizontal or vertical transitions cannot be considered as matches. Only entries along the path that result from diagonal transitions are considered as possible matches. Since each entry is a sphericity value, a small value signifies that these two landmarks do not match well locally with each other. Such an entry, if included as a match, will also introduce error in the estimation of the location of the object in the scene. We require that the entries along the path must be above a certain threshold before they can be considered as possible matches. A threshold value of 0.6 has been empirically developed. In the above example shown in Table II, entries (2, 11), (3, 12), (4, 1), (5, 2) are considered as possible matches. Isolated entries that have been considered as possible matches so far are then eliminated because they are not locally supported by their neighbors. At this point, entries along the path that are considered as matches must be sequences consisting of at least two consecutive diagonal entries. The example shown in Table II does not have any isolated entry, and hence entries considered as matches remain the same.

Since the sphericity value of each entry is derived from mapping a model landmark and its two adjacent landmarks to a scene landmark and its adjacent landmarks, a sphericity value that is close to one not only indicates that the model and the scene landmark match well locally with each other but also implies that their two respective adjacent landmarks match well with each other. The final step is to check the values of the entries that are considered as matches along the path. If the entry has a value that is greater than 0.9, its adjacent diagonal entries will also be considered as matches. In Table II, since all entries that are considered as matches between the model and the scene landmarks have sphericity value greater than 0.9, their respective adjacent diagonal entries are considered as matches. Thus, entries (1, 10) and (6, 3) are also considered as matches; they are adjacent to entries (2, 11) and (5, 2), respectively. In this example, model landmarks 1, 2, 3, 4, 5, and 6 match with scene landmarks 10, 11, 12, 1, 2, and 3, respectively.

B. Location Estimation and Matching Verification

After determining the landmarks of a model that match well with those in the scene using HDP, we next estimate the location of the model object in the scene, and verify the hypothesis that this model object is in the scene. Location of the object in the scene is estimated by finding a coordinate transformation consisting of translation, rotation, and scaling that maps the matched landmarks of the model to the corresponding scene landmarks in a least square sense. A score based on the least squared error of the mapping is used to quantify the overall goodness of the match between the model and the scene.

Continuing from the earlier example, the wire stripper is mapped into the scene, as shown in Fig. 10, by the least square coordinate transformation derived from the matched pairs of landmarks between the model and the scene. Note that if *a priori* knowledge of the scale of the object in the scene is available, the scale factor derived from the least square coordinate transformation can be used as an additional parameter for verifying the match.

The least squared errors only quantify how well a portion of the model landmarks match with the corresponding scene landmarks. A small error indicates that the portion of the model landmarks match well with the corresponding scene landmarks. It does not, however, account for the overall goodness of match. Denote ϵ as the least squared error derived from the matched pairs of landmarks between the model and the scene. To account for the overall goodness of the match between the model and the scene, we use the following heuristic measure which penalizes incomplete matching of the landmarks of the model:

$$\epsilon' = \begin{cases} \left(1.0 + \left(\frac{n-2}{k-2}\right) \log_2 \left(\frac{n-2}{k-2}\right)\right) \bar{\epsilon} & \text{for } k \geq 3, \\ \infty & \text{for } k = 0, 1, 2 \end{cases} \tag{9}$$

where

n = is the total number of landmarks of the model,
k = is the number of model landmarks that match the scene landmarks, and
$\bar{\epsilon} = \epsilon/(k \text{ (scale factor)})$, i.e., $\bar{\epsilon}$ is the normalized least squared error.

Note that $k \leq n$. The scale factor is derived from the coordinate transformation. The heuristic measure, ϵ', which can be regarded as the error measure for the overall goodness of match between the model and the scene, is referred to as the *match error*. If only one or two model landmarks match those in the scene, the least squared error is always zero because there always exists a coordinate transformation that perfectly maps a set of one or two points to another set. We consider such cases where two or less model landmarks match with those in the scene as undetermined cases; i.e., these cases have insufficient evidence of match between the model and the scene. The logarithmic term of the match error serves as a penalty factor for incomplete matching of the model landmarks. When all model landmarks match those in the scene ($k = n$), the match error equals the normalized least squared error. The penalty is larger when less model landmarks match those in the scene (k is smaller). In the earlier example, since all the model landmarks match those in the scene, the match error value of 0.62 is the same as the least squared error. The hypothesis of the model in the scene is finally determined by the value of the match error. With a small error we accept the hypothesis while with a large error we nullify the hypothesis. The decision strategy of the landmark-based shape recognition is thus a thresholding operation. If a match error is above a threshold, the match is considered correct; otherwise, the match

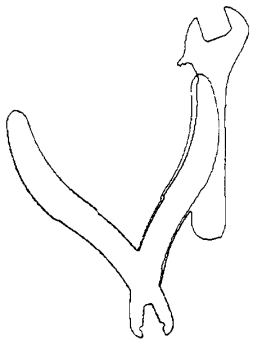


Fig. 10. The result of mapping the wire stripper into the scene shown in Fig. 9 by the least square coordinate transformation.

is considered incorrect. In our study, this threshold is set empirically.

C. Computational Complexity of Hopping Dynamic Programming

Let $N = \min(n, m)$ denote the minimum between the number of model landmarks and the number of scene landmarks. It can be deduced from the transition rules of HDP [18] that the computational complexity of HDP is bounded from above by

$$(N-1)^2 \text{ additions and } 2N(N-1) \text{ comparisons,}$$

and is bounded from below by

$0.5(N-1)^2$ additions and (N^2-1) comparisons when N is odd, or

$(0.5N^2 - N + 1)$ additions and N^2 comparisons when N is even.

As compared to [17] which has a complexity bounded between $O(n^4)$ and $O(n)$, the HDP has a complexity of $O(N^2)$.

The above computational complexity does not include any other overhead, such as switching between the forward and the backward procedure, looping, data transfer, etc. In general, as seen in the examples shown in Section III, each object is usually represented by less than 50 landmarks. It is thus computationally inexpensive to perform the landmark matching task by HDP.

VI. Experimental Results

In the following examples, landmarks in the scene are extracted based on the cardinal curvature points using $\omega = 20$. Consider again the scene shown in Fig. 9, the results of performing the landmark matching task between the scene and each of the tool models shown in Fig. 2 are summarized in Table III. Models that match well with the objects in the scene are those with the smallest match errors. Although the wire cutter is not in the scene, the match error between the wire cutter and the scene is quite small. The reason for this small match error is that the relative positions of the landmarks of the wire cutter are similar to those of the wire stripper which is in the scene.

Fig. 11 shows a more complicated scene which consists of six overlapping objects. Compared to their respective models, the specialty plier has been rotated by 20° and scaled by an area factor of 0.5. The wrench and Halmahera has been rotated by 90°; the spacecraft has been rotated by 180°. Luzon has been scaled by an area factor of 1.4, and Borneo has been rotated by 90° and scaled by an area factor of 0.6. Compared to their respective model landmarks, three out of six of the landmarks of the specialty plier, one out of six of the landmarks of the wrench, two out of seven of the landmarks of the spacecraft, two out of eight of the landmarks of Halmahera, five out of eighteen of the landmarks of Luzon, and three out of seven of the landmarks of Borneo are missing. With respect to each model, those landmarks in the scene not belonging

TABLE III
The Summary of the Results of Matching the Tool Models with the Scene Shown in Fig. 9. Identified Objects Are Indicated by "*".

Models	Model figure numbers	Total Number of Model landmarks	Number of matched model landmarks	Match Error
wrench *	2b	6	6	1.98
needle-nose plier	2d	4	2	∞
wire cutter	2e	6	5	7.39
specialty plier	2c	6	3	43.02
wire stripper *	2a	6	6	0.62

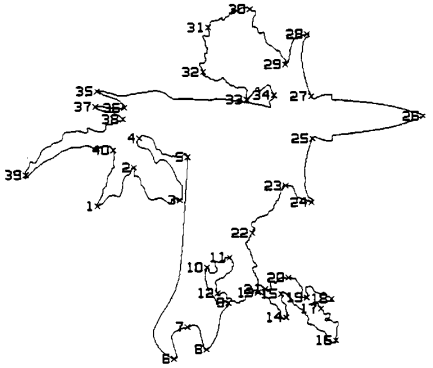


Fig. 11. A scene which consists of six overlapping objects—Borneo, Luzon, Halmahera, spacecraft, specialty plier, and wrench. Each landmark is labeled and indicated by an "×."

TABLE IV
The Summary of the Results of Matching the Library of Objects Shown in Fig. 2 with the Scene Shown in Fig. 11. Identified Objects Are Indicated by "*".

Models	Model figure numbers	Total Number of Model landmarks	Number of matched model landmarks	Match Error
wrench *	2b	6	4	0.74
needle-nose plier	2d	4	0	∞
wire cutter	2e	6	2	∞
specialty plier *	2c	6	3	7.89
wire stripper	2a	6	2	∞
Borneo *	2g	7	5	11.75
Halmahera *	2h	8	6	0.57
Luzon *	2i	18	14	0.78
Mindanao	2j	13	3	54.59
New Guinea	2k	11	5	127.74
Sulawesi	2l	9	4	18.08
spacecraft *	2f	7	5	0.55

to the model are considered as extraneous landmarks. The results of matching each model object of the library with the scene are summarized in Table IV. Again, the models that correctly match with the scene have the smallest match errors. Figs. 12–17 show the results of mapping Borneo, Luzon, Halmahera, spacecraft, specialty plier, and wrench into the scene, respectively.

Now consider a scene which consists of three overlapping objects, as shown in Fig. 18(a). The gray level value of the object region is 160, and the background is 96. Compared to their respective models, the needle-nose plier has been scaled by an area factor of 0.3, and the spacecraft has been rotated by 90° and scaled by an area factor of 0.6. The scene contour together with its landmarks is shown in Fig. 18(b). Although all the landmarks of the needle-noise plier appear in the scene, part of their sequential order is lost due to occlusion. Six out of the seven landmarks of the spacecraft appear in the scene, but only three (17, 18, 19) are in the correct sequential order. Nine out of thirteen landmarks of Mindanao are in correct sequential order. The results of matching each model object of the library with the scene are summarized in Table V.

To simulate the effects of noisy data, a zero mean i.i.d. Gaussian random variable is added to each pixel of the image. The noisy

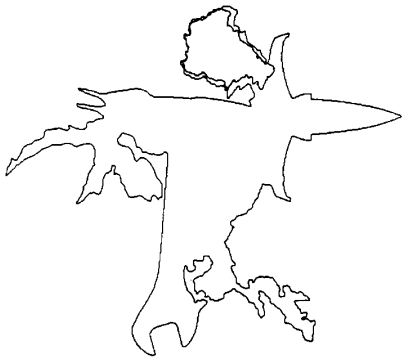


Fig. 12. The result of mapping Borneo into the scene shown in Fig. 11.

Fig. 13. The result of mapping Luzon into the scene shown in Fig. 11.

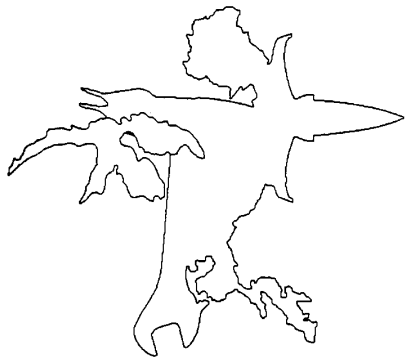


Fig. 14. The result of mapping Halmahera into the scene shown in Fig. 11.

Fig. 15. The result of mapping the spacecraft into the scene shown in Fig. 11.

Fig. 16. The result of mapping the specialty plier into the scene shown in Fig. 11.

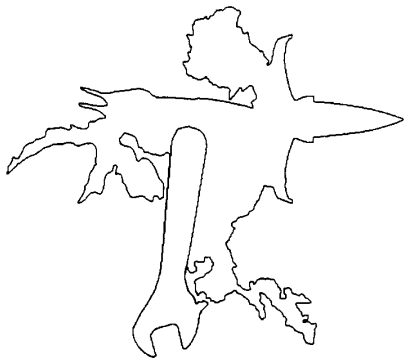


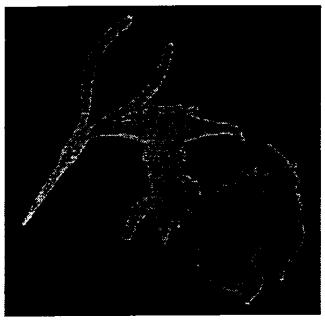
Fig. 17. The result of mapping the wrench into the scene shown in Fig. 11.

image is then thresholded at 128. The contours of the resulting regions in the thresholded image are traced, and the longest contour from which landmarks are extracted is used to represent the object contour in the noisy image. We will consider the noisy image as the image of the scene. Note that no attempt has been made to clean the noisy image.

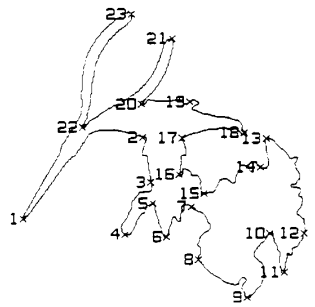
Denote σ as the standard deviation of the Gaussian random variables. The signal to noise ratio (SNR) of the noisy image is defined as:

$$\text{SNR} = 20 \log \frac{64}{\sigma} \text{ dB},$$

where 64 is the difference between the gray level values of the object region and the background. Fig. 19 shows the landmarks along the object contour extracted from the noisy image with a 3 dB SNR. The results of matching each model object of the library with the scene are summarized in Table VI. The sequential order of the landmarks in the scene compared to the respective model landmarks are lost. Only the match between Mindanao and the scene has a smaller error, and the rest are either mismatched or undetermined. Fig. 20 shows the landmarks along the object contour extracted from the noisy image with a 6 dB SNR. The exper-



(a)



(b)

Fig. 18. (a) A scene which consists of three overlapping objects—Mindanao, spacecraft, and needle-nose plier. (b) Extracted scene landmarks. Each landmark is labeled and indicated by an "×."

TABLE V
THE SUMMARY OF THE RESULTS OF MATCHING THE LIBRARY OF OBJECTS SHOWN IN FIG. 2 WITH THE SCENE SHOWN IN FIG. 18(b). IDENTIFIED OBJECTS ARE INDICATED BY "*".

Models	Model figure numbers	Total Number of Model landmarks	Number of matched model landmarks	Match Error
wrench	2b	6	3	77.11
needle-nose plier *	2d	4	4	0.24
wire cutter	2e	6	2	∞
specialty plier	2c	6	2	∞
wire stripper	2a	6	4	13.96
Borneo	2g	7	2	∞
Halmahera	2h	8	3	137.97
Luzon	2i	18	4	261.62
Mindanao *	2j	13	10	1.40
New Guinea	2k	11	2	∞
Sulawesi	2l	9	3	140.39
spacecraft *	2f	7	3	8.22

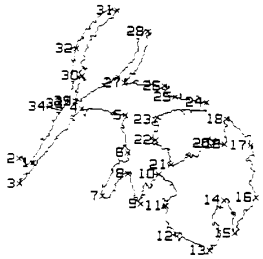


Fig. 19. Landmarks along the object contour extracted from the noisy image with SNR = 3 dB.

imental results are summarized in Table VII. When the sequential order of the landmarks in the scene are not severely changed compared to those of the respective models, matches are correctly determined yielding small match error values.

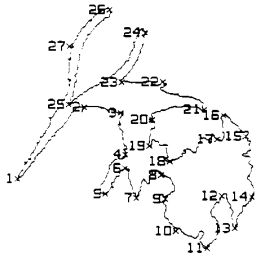


Fig. 20. Landmarks along the object contour extracted from the noisy image with SNR = 6 dB.

TABLE VI
THE SUMMARY OF THE RESULTS OF MATCHING THE LIBRARY OF OBJECTS SHOWN IN FIG. 2 WITH THE SCENE SHOWN IN FIG. 19. IDENTIFIED OBJECTS ARE INDICATED BY "*"

Models	Model figure numbers	Total Number of Model landmarks	Number of matched model landmarks	Match Error
wrench	2b	6	2	∞
needle-nose plier *	2d	4	0	∞
wire cutter	2e	6	0	∞
specialty plier	2c	6	0	∞
wire stripper	2a	6	3	78.17
Borneo	2g	7	3	14.42
Halmahera	2h	8	3	114.58
Luzon	2i	18	6	171.81
Mindanao *	2j	13	6	18.09
New Guinea	2k	11	3	116.42
Sulawesi	2l	9	4	53.22
spacecraft *	2f	7	4	16.08

TABLE VII
THE SUMMARY OF THE RESULTS OF MATCHING THE LIBRARY OF OBJECTS SHOWN IN FIG. 2 WITH THE SCENE SHOWN IN FIG. 20. IDENTIFIED OBJECTS ARE INDICATED BY "*"

Models	Model figure numbers	Total Number of Model landmarks	Number of matched model landmarks	Match Error
wrench	2b	6	2	∞
needle-nose plier *	2d	4	3	2.92
wire cutter	2e	6	3	62.56
specialty plier	2c	6	2	∞
wire stripper	2a	6	4	7.75
Borneo	2g	7	3	121.68
Halmahera	2h	8	3	158.96
Luzon	2i	18	5	159.37
Mindanao *	2j	13	6	14.47
New Guinea	2k	11	3	20.70
Sulawesi	2l	9	3	66.15
spacecraft *	2f	7	3	17.02

VII. CONCLUSIONS

The experimental results have demonstrated that the landmark matching task can handle occlusion reasonably well. It is difficult to theoretically analyze the performance of the landmark matching task which are, in many cases, problem dependent. The performance depends on the quality of the extracted scene landmarks, and the number of correct landmarks in a scene that are detectable. From Section IV, the match error is undefined if two or fewer landmarks of a model are correctly matched with the scene landmarks. Therefore, when matching landmarks of a model with those in a scene, at least three landmarks in a scene that correspond to the model must be detectable. In addition, part of the sequential order of the detectable landmarks must be preserved. From the experimental results, it is safe to say that an object in a scene can be recognized as long as more than half of its landmarks in the scene can be detected in the correct sequential order. It is also important to note that distortion in landmark locations does not degrade the matching performance as much as distortion in the sequential order of the landmarks.

REFERENCES

[1] F. Attneave, "Some informational aspects of visual perception," *Psychol. Rev.*, vol. 61, no. 3, pp. 183–193, 1954.

[2] F. L. Bookstein, *The Measurement of Biological Shape and Shape Change*. New York: Springer-Verlag, 1978.
[3] R. C. Bolles and R. A. Cain, "Recognizing and locating partially visible objects: The local-feature-focus method," *Int. J. Robotics Res.*, vol. 1, no. 3, pp. 57-82, Fall 1982.
[4] M. W. Koch and R. L. Kashyap, "Using polygons to recognize and locate partially occluded objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, no. 4, pp. 483-494, July 1987.
[5] D. P. Huttenlocher and S. Ullman, "Object recognition using alignment," in *Proc. IEEE 1st Int. Conf. Computer Vision*, London, 1987, pp. 102-111.
[6] L. S. Davis, "Shape matching using relaxation techniques," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-1, no. 1, pp. 60-72, Jan. 1979.
[7] B. Bhanu and O. D. Faugeras, "Shape matching of two-dimensional objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, no. 2, pp. 137-155, Mar. 1984.
[8] K. E. Price, "Matching closed contours," in *Proc. Seventh Int. Conf. Pattern Recognition*, Montreal, P.Q., Canada, July 30-Aug. 2, 1984, pp. 990-992.
[9] N. Ayache and O. D. Faugeras, "HYPER: A new approach for the recognition and positioning of two-dimensional objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, no. 1, pp. 44-54, Jan. 1986.
[10] B. Bhanu and J. C. Ming, "Recognition of occluded objects: A cluster-structure algorithm," *Pattern Recogn.*, vol. 20, no. 2, pp. 199-211, 1987.
[11] J. L. Turney, T. N. Mudge, and R. A. Volz, "Recognizing partially occluded parts," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-7, no. 4, pp. 410-421, July 1985.
[12] T. F. Knoll and R. C. Jain, "Recognizing partially visible objects using feature indexed hypotheses," *IEEE Trans. Robotics and Automation*, vol. RA-2, no. 1, pp. 3-13, Mar. 1986.
[13] A. Kalvin, E. Schonberg, J. T. Schwartz, and M. Sharir, "Two-dimensional model-based, boundary matching using footprints," *Int. J. Robotics Res.*, vol. 5, no. 4, pp. 38-55, Winter 1986.
[14] G. J. Ettinger, "Large hierarchical object recognition using libraries of parameterized model sub-parts," in *Proc. IEEE Comput. Soc. Conf. Computer Vision and Pattern Recognition*, Ann Arbor, MI, June 5-9, 1988, pp. 32-41.
[15] W. E. L. Grimson, "On recognition of curved objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 6, pp. 632-643, June 1989.
[16] Y. Lamdan, J. T. Schwartz, and H. J. Wolfson, "Object recognition by affine invariant matching," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Ann Arbor, MI, June 5-9, 1988, pp. 335-344.
[17] ——, "On recognition of 3-D objects from 2-D images," in *Proc. IEEE Int. Conf. Robotics and Automation*, Philadelphia, PA, Apr. 1988, pp. 1407-1413.
[18] N. Ansari, "Shape recognition: A landmark-based approach," Ph.D. dissertation, School Elec. Eng., Purdue Univ., West Lafayette, IN, Aug. 1988; also issued as School Elec. Eng., Purdue Univ., Tech. Rep. TR-EE-88-31, July 1988.
[19] J. W. Gorman, O. R. Mitchell, and F. P. Kuhl, "Partial shape recognition using dynamic programming," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, no. 2, pp. 257-266, Mar. 1988.
[20] W. E. L. Grimson and T. Lozano-Perez, "Model-based recognition and localization from sparse range or tactile data," *Int. J. Robotics Res.*, vol. 3, no. 3, pp. 3-35, Fall 1984.
[21] P. J. Besl and R. C. Jain, "Three-dimensional object recognition," *ACM Comput. Surveys*, vol. 17, no. 1, pp. 75-154, 1985.
[22] V. Torre and T. A. Poggio, "On edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, no. 2, pp. 147-163, Mar. 1986.
[23] H. Asada and M. Brady, "The curvature primal sketch," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, no. 1, pp. 2-14, Jan. 1986.
[24] D. Gans, *Transformations and Geometries*. New York: Appleton-Century-Crofts, 1969.
[25] R. G. Bartle, *The Elements of Real Analysis*. New York: Wiley, 1976.
[26] R. J. Muirhead, *Aspects of Multivariate Statistical Theory*. New York: Wiley, 1982.
[27] G. A. F. Seber, "The non-central chi-squared and beta distributions," *Biometrika*, vol. 50, pp. 542-544, 1963.
[28] B. O'Neill, *Elementary Differential Geometry*. New York: Academic, 1966.
[29] S. E. Dreyfus and A. M. Law, *The Art and Theory of Dynamic Programming*. New York: Academic, 1977.

On Curve Matching

HAIM J. WOLFSON

***Abstract*—Two algorithms to find the longest common subcurve of two 2-D curves are presented. These algorithms are based on conversion of the curves into *shape signature* strings and application of string matching techniques to find long matching substrings. Then direct curve matching is applied to the corresponding 'candidate' subcurves to find the longest matching subcurve. The first algorithm is of complexity $O(n)$, where n is the number of sample points on the curves. The second one, while being theoretically somewhat less efficient, proved to be robust and efficient in practical applications. Both algorithms solve the problem for general curves without being dependent on some set of *special points* on the curves. The algorithms have industrial applications to problems of object assembly and object recognition. Experimental results are included. The algorithms can be easily extended to the 3-D case.**

***Index Terms*—Computer vision, curvature, curve matching, machine intelligence, object recognition, part assembly, pattern recognition, string matching.**

I. Introduction

The problem of finding the best fit between two curves is of central importance in robotic applications of computer vision. This problem appears in various assembly tasks requiring a robot to put two pieces together along their matching boundary, e.g., "puzzle assembly" [1]-[3]. Another major application for such an algorithm is in recognition and location of partially occluded objects in an overlapping scene [4]-[6]. Since two-dimensional objects are completely described, both globally and locally, by their closed boundary curves, the detection of partially occluded objects participating in a composite scene can be done by matching the boundary curve of the scene with the boundary curves of *candidate* objects, and trying to find out whether they have a "long enough" matching subcurve. This method is particularly attractive in recognition of partially occluded objects, because in such a situation no use can come of global object characteristics, and only properties which are preserved locally, such as the visible boundary curve, can be taken into account. The method can also be applied to three-dimensional objects which are flat enough, or have a small number of stable positions, so that the orthogonal projection of each such position can be taken as a 2-D model. Curve matching algorithms also have potential applications in finding correspondence between maps and terrain images. An earlier version of *Algorithm II* of this

Manuscript received July 29, 1988; revised September 11, 1989. This work was supported by Office of Naval Research under Grant N00014-82-K-0381, the National Science Foundation under Grant NSF-DCR-83-20085, and by grants from Digital Equipment Corporation and IBM Corporation.

The author is with the Robotics Research Laboratory, Courant Institute of Mathematical Sciences, New York University, 715 Broadway, 12th floor, New York, NY 10003, and the Department of Computer Science, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv 69978, Israel.

IEEE Log Number 8933762.

0162-8828/90/0500-0483$01.00 © 1990 IEEE