PURDUE UNIVERSITY

Graduate School

This is to certify that the thesis prepared

By Nirwan Ansari

Entitled Shape Recognition: A Landmark-Based Approach

Complies with University regulations and meets the standards of the Graduate School for originality and quality

For the degree of Doctor of Philosophy

Signed by the final examining committee:

, chair

Approved by the head of school or department:

July 29 1988

This thesis ☐ is ☒ is not to be regarded as confidential

Major professor

Grad. School
Form No. 9
Revised 9-85

SHAPE RECOGNITION : A LANDMARK-BASED APPROACH

A Thesis

Submitted to the Faculty

of

Purdue University

by

Nirwan Ansari

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 1988

to my parents
for their perpetual love and support

ACKNOWLEDGMENTS

The author wishes to express his sincere appreciation and gratitude to his major professor, Professor Edward J. Delp, for his continued support, guidance, and encouragement during the years of the author's graduate studies at both Purdue University and the University of Michigan.

Appreciation is also extended to the author's other committee members, Professor Jan P. Allebach, Professor Lawrence L. Ogborn, Professor Allen W. Weitsman, and Professor Michael D. Zoltowski, for serving on the examining committee and for evaluating this research.

For the introduction to the theory of quasiconformal mappings the author is indebted to Professor Allen W. Weitsman. A special thanks also goes to the author's colleagues, Chee-Hung Chu and Hin-Leong Tan, for their comments on this manuscript.

Financial assistance provided by Purdue University and IBM while at Purdue is greatly appreciated.

Finally, the author wishes to thank his wife, Chun-Lei, for her patience and constant support.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table

Page

LIST OF FIGURES

Figure

Page

Figure

Page

Figure

Page

Figure Page

Figure

Page

Figure

Page

Figure Page

Figure

Page

ABSTRACT

Ansari, Nirwan. Ph.D., Purdue University. August 1988. Shape Recognition: A Landmark-Based Approach. Major Professor: Edward J. Delp.

Shape recognition has applications in computer vision tasks such as industrial automated inspection and automatic target recognition. When objects are occluded, many 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 which have important shape attributes and are usually obtained from the object boundary. In this study, we use high curvature points along an object boundary as the landmarks of the object. Given a scene consisting of partially occluded objects, the hypothesis of a model object in the scene is verified by matching the landmarks of an object with those in the scene. A measure of similarity between two landmarks, one from a model and the other from a scene, is needed to perform this matching. One such local shape measure is the sphericity of a triangular transformation mapping the model landmark and its two neighboring landmarks to the scene landmark and its two neighboring landmarks.

Sphericity is in general defined for a diffeomorphism. Its invariant properties under a group of transformation, namely, translation, rotation, and scaling are derived. The sphericity of a triangular transformation is shown to

be a robust local shape measure in the sense that minor distortion in the landmarks does not significantly alter its value.

To match landmarks between a model and a scene, a table of compatibility, where each entry of the table is the sphericity value derived from the mapping of a model landmark to a scene landmark, is constructed. A hopping dynamic programming procedure which switches between a forward and a backward dynamic programming procedure is applied 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 heuristic measure is then computed to decide if the model is in the scene.

CHAPTER 1

INTRODUCTION

1.1. Shape

A tremendous amount of research has been undertaken to understand how the human visual system functions. The present state of the art computer vision systems are still a long way off from being able to mimic trivial visual tasks that human beings perform routinely. Shape recognition is one such task that remains a difficult computer vision problem.

The visual perception of an object is determined by many factors such as luminance, chromaticity, contrast, acuity, texture, and the shape of the object. We shall use the word *shape* to refer to the invariant geometrical properties of the relative distances among a set of static spatial features of the object. These static spatial features are known as the *shape features* of the object. Two objects are said to have the same shape if there exists a similarity transformation, which consists of a combination of translation, rotation, and scaling, that maps the shape features of one object into those of the other object.

1.2. A General Shape Recognition System

The computer vision task of shape recognition is that of identifying specific objects in one or more images of a scene. The images may be acquired from sensors that respond to one of several possible physical stimuli such as light, heat, or motion. The sensor data are usually arranged in a discrete array. Each element of an (image) array is known as a pixel, and it represents the energy of a physical stimulus in a particular area. For example, an infrared image corresponds to the thermal energy of a scene. Though other imageries are equally applicable to the shape recognition task, we shall only consider gray level intensity images.

A general computer shape recognition system using a single view is illustrated in Figure 1.1. Given an image of a scene, the system tries to recognize and identify the objects in the scene by mapping them to models stored in a library. The system consists of three functional blocks — feature

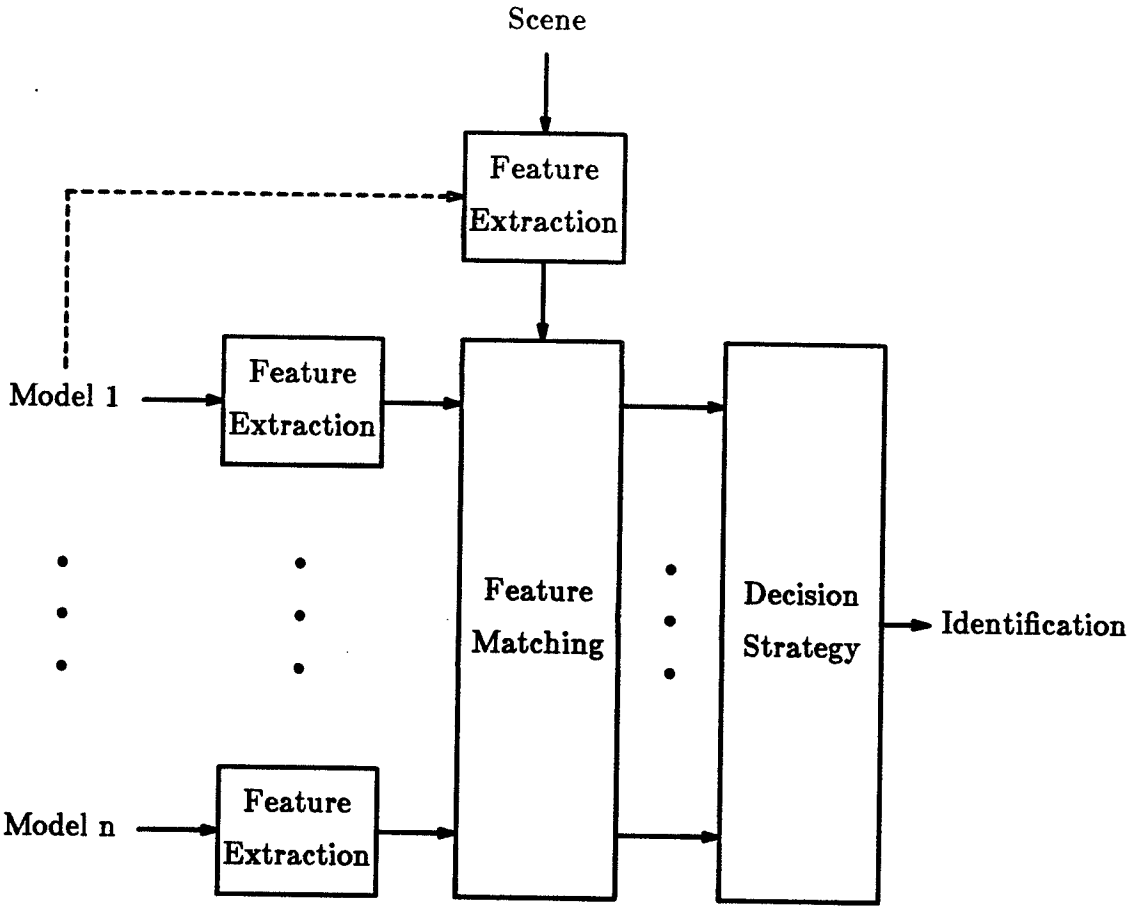


Figure 1.1 A general shape recognition system.

extraction, feature matching, and a decision strategy. Each model of the object is represented by its associated shape features which are extracted in advance and stored in the library. The complexity of the feature extractor depends on the kind of shape features to be detected. Model features can be extracted in a well controlled lighting condition. They can also be derived from computer aided design (*CAD*) models, or chosen based on *a priori* knowledge. In a bottom-up (data-driven) approach, shape features of objects in the scene are extracted from raw image data without knowledge of the geometric structures or other visual properties of the models. Extracted shape features of objects in the scene are then compared to those of each model. Based on a decision strategy on how well the shape features of each model are matched to those in the scene, objects in the scene are identified. If knowledge of what is to be expected in a scene and a high level description of each model is available, a top-down (goal-directed) approach can be used to aid the shape recognition task. The shape features of a model usually form a high level description of the model, such as its geometric structure. Knowledge of each model is used to guide extracting shape features in the scene. Again, based on a decision strategy on how well shape features of each model are matched to those in the scene, objects in the scene are identified.

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 object. After extracting the shape features from a model and a scene, some kind of similarity or dissimilarity measures must be used to quantify the difference between the shape features. These similarity or dissimilarity measures are referred to as ***shape measures.*** Since the shape of an object refers to the invariant geometrical properties about the relative distances among the shape features of the object, it should remain the same when the object is viewed at a different scale or 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 or dissimilarity between two entire objects; a ***local shape measure*** quantifies the similarity or dissimilarity 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. A shape recognition method that uses global shape features and global shape measures to

achieve recognition is known as a *global shape recognition method*; one that uses local shape features and local shape measures is known as a *local shape recognition method.*

1.3. Problem Statement

Extensive research work on two-dimensional (*2-D*) global shape recognition has been done in the past two decades. Applications include recognition of handwritten characters, biological cells, and industrial parts. Surveys of various 2-D global shape recognition methods such as Fourier shape descriptors, moment shape descriptors, and template matching can be found in [Pav78, Pav80, Lev85, Ott88]. These approaches are applicable only when the entire object contour or silhouette is available.

The problem we want to address is that of recognizing and locating planar objects that may be occluded or touching each other. A typical situation is shown in Figure 1.2, where there are three objects occluding each other. These objects are almost flat with one dimension being much smaller than the other two. Our task in this situation is to identify and locate the three individual objects in the scene. Since entire object contours or silhouettes are not available, global shape recognition methods will fail to identify such partially occluded objects.

1.4. Landmark-Based Approach

Recent work [Bol82, Bha84, Pri84, Bha87, Aya86, Koc87, Tur85, Kno86, Gor88] on 2-D partial shape recognition have exhibited an increasing interest in developing methods capable of recognizing objects when global information about the objects are not available. Most of the approaches use as their shape features line segments resulting from a polygonal approximation to the object contour. They will all be reviewed in Chapter 4.

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 [Att54] 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 [Boo78] to study and observe the growth of biological objects. One such application is the study of craniofacial growth [Boo84] by observing the changes of dominant points of a cranial face at two time intervals. Cardiac images can also be analyzed by observing the changes of some dominant points along the cardiac wall in a sequence of echocardiograms [Boo85, Sko86].

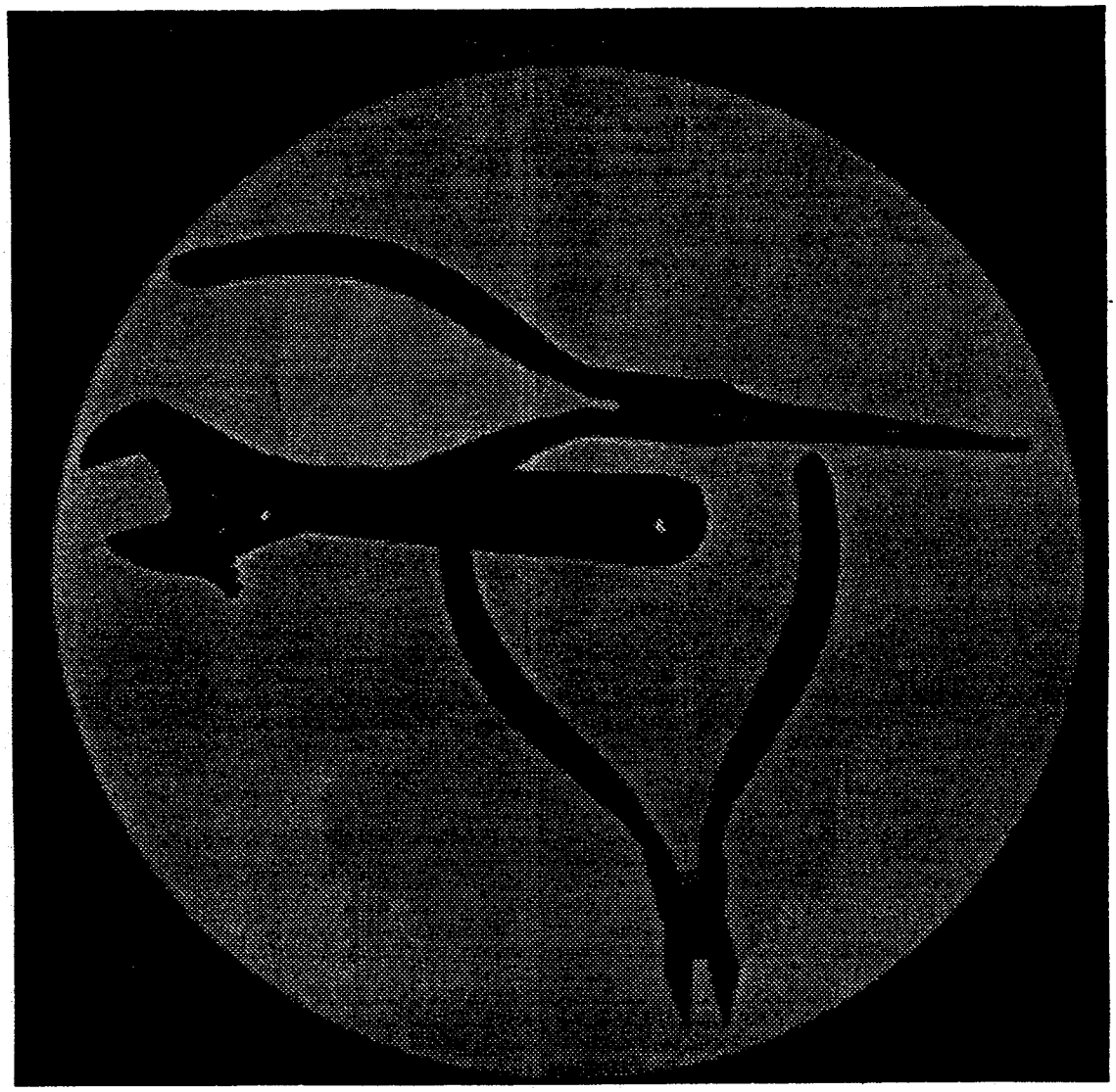


Figure 1.2. A typical scene consisting of occluding objects.

These dominant points of an object are usually referred to as the landmarks of the object. However, we shall 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 extreme 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. Figure 1.3 shows the potential landmarks of an object. The *landmark-based shape recognition* approach that we shall present 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. If all the landmarks of an object in the scene are available, global recognition of the object can be achieved. If only a portion of the landmarks of the object are available, the identity of the object may be deduced by recognizing a portion of the object.

A landmark-based shape recognition system is shown in Figure 1.4. It is similar to the general shape recognition system shown in Figure 1.1. Landmarks extracted from a model object and from the scene are referred to as *model landmarks* and *scene landmarks*, respectively. Properties of landmarks of each model can be used to guide the extraction of landmarks in the scene. The hypothesis of a model object in the scene is made by matching the model landmarks to the scene landmarks. Based on a decision strategy on how well the landmarks of each model are matched to those of the scene, objects in the scene are identified.

1.5. The Scope of Work

In the remaining chapters, we shall discuss in detail all the functional blocks shown in Figure 1.4. We shall define two local shape measures known as dilatation and sphericity in Chapter 2. Both measures can be used to quantify the similarity or dissimilarity between a model landmark and a scene landmark. Their invariant properties will also be derived. Sphericity will be shown to be a robust local shape measure in the sense that a small perturbation in the landmark locations does not significantly alter its value. The landmark-based shape recognition approach does not require the extraction of the entire object contour. However, for illustrative purposes, we shall discuss two data-driven landmark extraction methods in Chapter 3. We shall only consider one type of

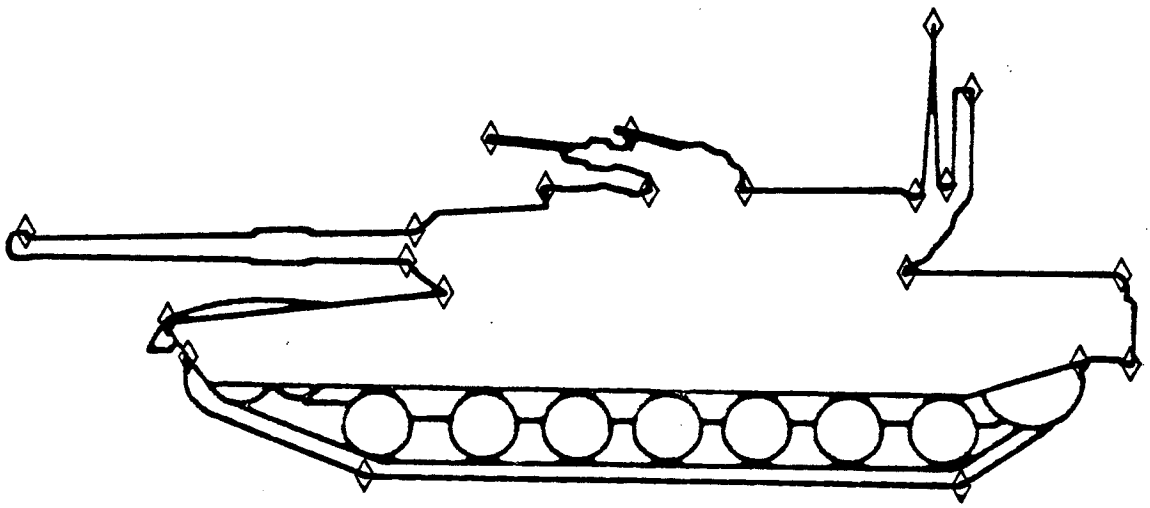


Figure 1.3. Potential landmarks of a tank. Each landmark is indicated by a diamond symbol.

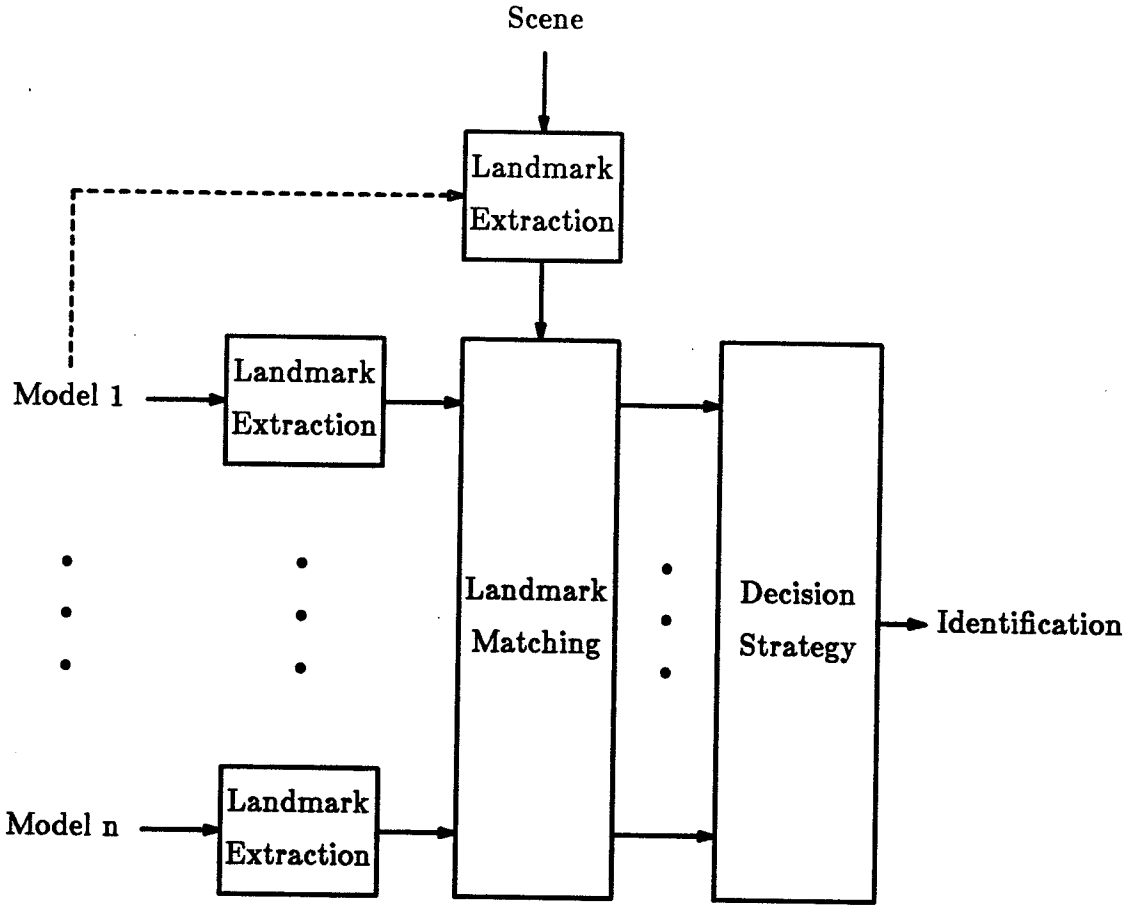


Figure 1.4. A landmark-based shape recognition system.

landmark — extreme curvature points along object contour. The object contour can be the boundary of an object region which may consist of more than one object overlapping each other. Other problem specific types of landmarks will not be considered. The first landmark extraction method is known as curvature guided polygonal approximation. It uses extreme curvature points along an object contour as an initial starting set of break points for a subsequent polygonal approximation algorithm. The subsequent polygonal approximation algorithm is a split-and-merge algorithm similar to the one described in [Pav74]. The second method is based on the cardinal curvature points along an object contour. The cardinal curvature points are obtained from successive Gaussian smoothing of a contour. In Chapter 4, we shall review the recent literature on 2-D partial shape recognition methods, and compare them with our landmark-based approach. We shall use landmarks as shape features and sphericity as a shape measure. Hypothesis of matches between model landmarks and scene landmarks is made by a HOPPING dynamic programming procedure. Final matches as well as the location of the object in the scene are presented. In Chapter 5, we shall present further experimental results. We shall conclude and present future directions in Chapter 6.

CHAPTER 2

MATHEMATICAL PRELIMINARIES: THE PROPERTIES OF A LOCAL SHAPE MEASURE

2.1. Introduction

All shape recognition methods employ some kind of shape measure, whether global or local, *ad hoc* or robust, derived from features that are extracted from different shapes to achieve recognition. In landmark-based shape recognition, as discussed in Chapter 1, we represent each object by a set of landmarks. To determine if a model object matches a scene object one must determine how well their corresponding landmarks match each other. Since each object is approximated by its associated landmarks with a polygon, matching landmarks of a model object with those in the scene becomes the problem of matching vertices of the two polygons associated with the model and the scene, respectively. A measure of similarity between two vertices, one from a model polygon and the other from a scene polygon, is needed to achieve this matching. One such local shape measure is the sphericity of a *triangular transformation* which maps one triangle into another triangle.

This chapter justifies the use of the sphericity of a triangular transformation as a local shape measure. Before describing the sphericity, we shall first define a related local shape measure known as the dilatation of a triangular transformation. The dilatation of a triangular transformation is also known as the anisotropy from the work of Bookstein [Boo78, Boo84] in the field of morphometrics. We have adopted the term, dilatation, that is used in the mathematics community. Bookstein [Boo84] computes the dilatations derived from various manually chosen points of a cranial outline at two time intervals to study craniofacial growth. However, he does not show how the dilatation of a triangular transformation is computed. It will be shown that the dilatation can be computed by means of three different mathematical approaches: direct geometric method, quasiconformal mapping, and through the use of strain tensors. The dilatation is not only defined for a triangular transformation, but

it will also be defined for a diffeomorphism.[1] In Section 2.3, we shall derive the invariant properties of the dilatation. Sphericity will be introduced in Section 2.4 and will be shown to be more mathematically tractable. Its invariant properties will also be derived. The sphericity takes on values in [0, 1] while the dilatation takes on values in [1, ∞). In Section 2.5, we shall show that the sphericity of a triangular transformation is a robust local shape measure. That is, minor distortion in the landmark locations (vertices of a triangle) does not significantly change the value of the sphericity. We assume that the distortion in the scene landmarks can be modeled as "noise" added to the model landmark locations. The probability density function of the sphericity of a triangular transformation will then be derived. In Section 2.6, we shall empirically estimate the probability density function of the sphericity for cases where the probability density function cannot be obtained in closed form.

To investigate the properties of dilatation and sphericity, various mathematical principles will be used. Existing definitions and theorems which are used to derive the desired properties of these local shape measures will be stated without proof.

2.2. Dilatation

The dilatation of a triangular transformation that maps one triangle to another triangle is a measure of similarity between the two triangles. Under such a triangular transformation, the inscribed circle of a triangle is mapped onto an inscribed ellipse of the other triangle. The *dilatation* of the triangular transformation is the ratio of the length of the major axis to that of the minor axis of the inscribed ellipse. This is shown in Figure 2.1, where the dilatation $= \frac{d_1}{d_2}$. Note that if the two triangles are similar, the dilatation is 1. As described below, the triangular transformation is uniquely determined by an affine transformation. The dilatation of a triangular transformation can be evaluated by a direct geometric approach. It can also be evaluated by means of a quasiconformal mapping and the strain tensor of an affine transformation.

[1] A *diffeomorphism* is a continuous one-to-one mapping whose inverse mapping is also continuous, with both the mapping and its inverse having continuous partial derivatives.

2.2.1. Affine Transformation

Definition 2.1: An affine transformation is a mapping of $\mathbf{x}$ to $\mathbf{u}$, where, $\mathbf{x}, \mathbf{u} \in \mathbb{R}^2$, defined by:

$$\mathbf{u} = A\mathbf{x} + \mathbf{t} \ , \tag{2.1}$$

where

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} , \ \mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix} , \ \mathbf{t} = \begin{bmatrix} e \\ f \end{bmatrix} ,$$

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} , \text{ and } \ det(\ A\ ) \neq 0 \ .$$

An affine transformation is a one-to-one mapping of the plane onto itself with the following properties [Gan69] :

- Collinearity is preserved under the transformation.
- Noncollinearity is preserved under the transformation.
- Betweenness relation is preserved under the transformation, i.e., a mid-point of a line remains the mid-point of the transformed line.

Thus, a point is mapped into a point, a line into a line, and an angle remains an angle. The six coefficients which describe an affine transformation are uniquely determined if three noncollinear points and their corresponding non-collinear image points are known. The term *image point*[2] used in this chapter is in the mathematical sense. It is not an image pixel.

By direct substitution of the equation of a circle into an affine transformation, it is readily seen that the transformed circle is an ellipse. It can also be shown that the area of the triangle that results from an affine transformation is scaled by det(A). Coefficients of the unique affine transformation which maps one triangle into another are computed using the following equation, (see Figure 2.1):

$$\begin{bmatrix} a \\ b \\ e \end{bmatrix} = B^{-1} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \qquad \text{and} \qquad \begin{bmatrix} c \\ d \\ f \end{bmatrix} = B^{-1} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} , \tag{2.2}$$

[2] Let $A \subset \mathbb{R}^2$ and $B \subset \mathbb{R}^2$ be two sets and f be a mapping of A into B. If $x \in A$, $f(x)$ is defined as the *image point* of x.

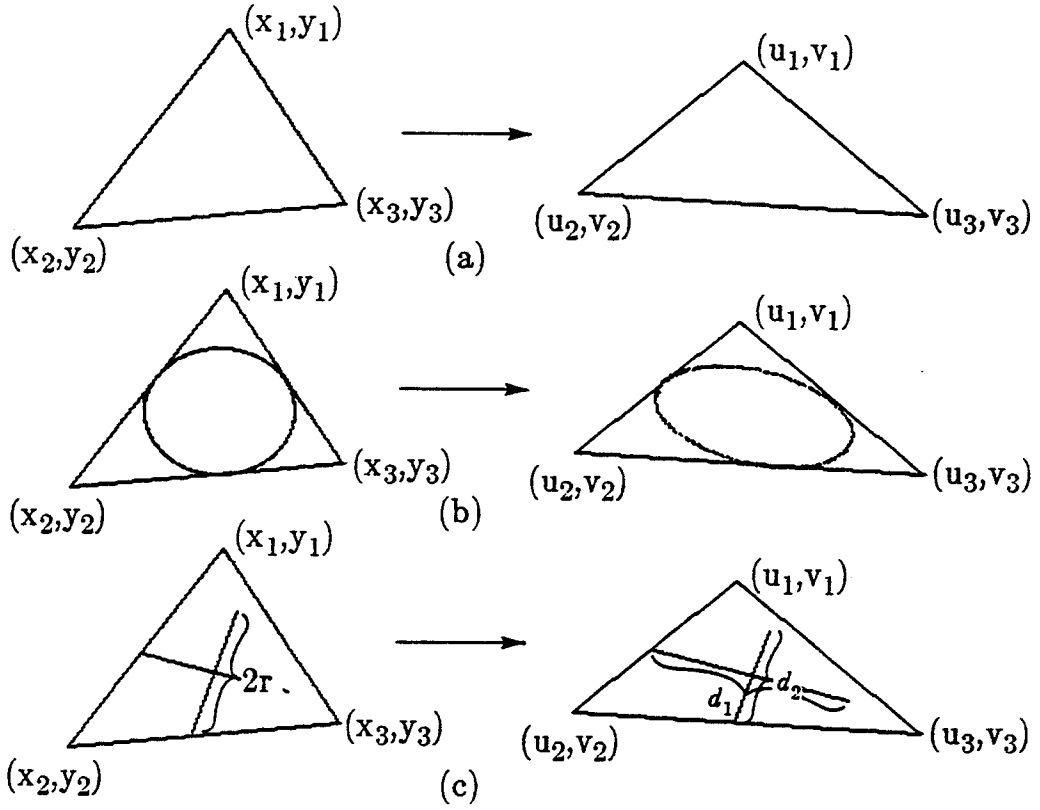


Figure 2.1. Representation of a mapping from a triangle to another triangle. (a) Original triangles. (b) Mapping from the inscribed circle to an inscribed ellipse. (c) Mapping of the principal axes.

$$\text{dilatation} = \frac{d_1}{d_2}, \qquad \text{sphericity} = \frac{2\sqrt{d_1 d_2}}{d_1+d_2}.$$

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

(u_i, v_i) are the image points of the points (x_i, y_i), i=1,2,3 under the transformation described by Equation 2.1. Since the 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 (radius and center) of the inscribed circle. The inscribed circle is then mapped through the transformation into an ellipse from which the parameters of the ellipse can be obtained; and thus the dilatation of the triangular transformation can be computed. This is the direct geometric approach.

2.2.2. Quasiconformal Mapping

Quasiconformal mapping was introduced over half of a century ago, and in recent years, there has been a great deal of work in the area. We shall use these results to find the dilatation described in the previous section.

2.2.2.1. Two-Dimensional Quasiconformal Mapping

Quasiconformal mapping in the plane has geometrical properties similar to the triangular transformation mentioned above. The following is one definition, due to Ahlfors [Ahl66], used for a two-dimensional quasiconformal mapping.

Definition 2.2: Let $w = g(z)$, $w, z \in \mathbb{C}$ $(w = u + iv, z = x + iy)$, be a complex valued differential mapping, such that the differential

$$dg = g_z dz + g_{\bar{z}} d\bar{z} ,$$

where $\bar{z}$ is the complex conjugate of z,
g_z and $g_{\bar{z}}$ are the partial derivatives of g with respect to z and $\bar{z}$, respectively.

Note that dg maps a circle around z into an ellipse around $g(z)$. Consider the case that g is sense preserving (the Jacobian is positive), then $(|g_z| - |g_{\bar{z}}|)|dz| \leq |dg| \leq (|g_z| + |g_{\bar{z}}|)|dz|$. g is said to be K-quasiconformal for some constant K if the ratio of the length of the

major axis to the length of the minor axis of the ellipse is

$$D_g(z) = \frac{|g_z| + |g_{\bar{z}}|}{|g_z| - |g_{\bar{z}}|} \leq K < \infty \,. \tag{2.3}$$

$D_g(z)$ is the dilatation of the mapping at z.

For notational convenience, the argument of the dilatation may be omitted. Note that the dilatation takes on values in $[1, \infty)$. The mapping is conformal if the dilatation equals 1. The following lemma follows immediately from the above definition.

Lemma 2.3: If g is an affine transformation defined by Equation 2.1 and $|A| > 0$, then D_g , the dilatation, is a constant and

$$D_g = \frac{\sqrt{g_D} + 1}{\sqrt{g_D} - 1} \quad \text{for} \quad g_D > 1 \,, \tag{2.4}$$

where

$$g_D = \frac{(a+d)^2 + (c-b)^2}{(a-d)^2 + (c+b)^2} \,.$$

Proof: The lemma is obtained by direct substitution of Equation 2.1 into Definition 2.2. □

This provides a simple computational formula for obtaining the dilatation of the above triangular transformation. The dilatation of a triangular transformation is 1 if the two triangles are similar. The less similar the triangles, the larger the value of the dilatation. We shall call g_D the ***conformity*** of the triangular transformation. The conformity is more computationally efficient than the dilatation. It can also be used as a shape measure. It is seen from Equation 2.4 that the larger the conformity is, the smaller the dilatation is, and vice versa. Like the dilatation, the conformity takes on values in $[1, \infty)$. The probability density function of the conformity will be discussed in Section 2.5.

2.2.2.2. n-Dimensional Quasiconformal Mapping

In this section, we shall define the dilatation of a diffeomorphism. A ***diffeomorphism*** is a continuous one-to-one mapping whose inverse mapping is also continuous, with both the mapping and its inverse having continuous

partial derivatives [O'N66]. The following definition of n-dimensional quasiconformal mapping is due to Vaisala [Vai61]:

Definition 2.4: A diffeomorphism, $g:\Omega\rightarrow\overline{\Omega}$, $(\Omega, \overline{\Omega} \subset \mathbb{R}^n)$ is called K-quasiconformal if

$$\sup_{\mathbf{x}\in\Omega} D_g(\mathbf{x}) \leq K < \infty \quad \text{for some constant } K, \tag{2.5}$$

where $D_g(\mathbf{x}) = \max(\overline{D}_g(\mathbf{x}), \underline{D}_g(\mathbf{x}))$,

$$\overline{D}_g(\mathbf{x}) = \frac{(\max_{|\mathbf{e}|=1} |g'(\mathbf{x})\mathbf{e}|)^n}{|J(\mathbf{x})|}, \quad \underline{D}_g(\mathbf{x}) = \frac{|J(\mathbf{x})|}{(\min_{|\mathbf{e}|=1} |g'(\mathbf{x})\mathbf{e}|)^n}, \text{ and}$$

$\mathbf{e}$ is an unit vector in $\mathbb{R}^n$.

$J(\mathbf{x})$ and $g'(\mathbf{x})$ are the Jacobian and the derivative (Jacobian matrix) of $g(\mathbf{x})$, respectively. $\underline{D}_g(\mathbf{x})$, $\overline{D}_g(\mathbf{x})$, and $D_g(\mathbf{x})$ are known as the inner, outer, and maximal dilatation at $\mathbf{x}$, respectively.

For notational convenience, the arguments of the various function used in Definition 2.4 may be omitted. The following lemmas, Lemmas 2.5-2.7, relate the dilatations to the eigenvalues of $g'^t g'$. Note that g' maps a unit ball in Ω into an ellipsoid in $\overline{\Omega}$. The eigenvalues of $g'^t g'$ correspond to the lengths of the semi-axes of the transformed ellipsoid. Since $J(\mathbf{x}) \neq 0$, $g'^t g'$ is positive definite [Seb77]. The eigenvalues of $g'^t g'$ are thus positive.

Lemma 2.5: $\max_{|\mathbf{e}|=1} |g'(\mathbf{x})\mathbf{e}|^2$ corresponds to the largest eigenvalue of $g'^t g'$.

Proof: $|g'\mathbf{e}|^2 = < g'\mathbf{e}, g'\mathbf{e} >$

$$= < g'^t g'\mathbf{e}, \mathbf{e} >$$

$$= < \Phi^t D \Phi \mathbf{e}, \mathbf{e} >$$

$$= < D\tilde{\mathbf{e}}, \tilde{\mathbf{e}} >$$

$$= \lambda_1 \tilde{e}_1^2 + \cdots + \lambda_n \tilde{e}_n^2,$$

where $\Phi^t D \Phi$ is the spectral decomposition [Fri79] of $g'^t g'$, and

$\lambda_n \geq \lambda_{n-1} \geq \cdots \geq \lambda_1 > 0$, are the eigenvalues of $g'^t g'$.

Since $\tilde{\mathbf{e}} = \Phi \mathbf{e} = (\tilde{e_1}, \cdots, \tilde{e_n})^t$, $|\mathbf{e}| = 1$ implies $|\tilde{\mathbf{e}}| = 1$.

Therefore, $\max_{|\mathbf{e}|=1} |g'(\mathbf{x})\mathbf{e}|^2 = \lambda_n$. □

Lemma 2.6: $\min_{|\mathbf{e}|=1} |g'(\mathbf{x})\mathbf{e}|^2$ corresponds to the smallest eigenvalue of $g'^t g'$.

Proof: The proof is similar to the above lemma. □

Lemma 2.7: $J^2(\mathbf{x})$ is the product of the eigenvalues of $g'^t g'$.

Proof: $J^2(\mathbf{x}) = \det^2(g')$

$= det(g'^t g') = \lambda_1 \cdots \lambda_n$,

where $det()$ is the matrix determinant. □

Now, the dilatations can all be expressed in terms of the eigenvalues of $g'^t g'$ by the following:

$$\bar{D}_g = \left(\frac{{\lambda_n}^n}{|\lambda_1 \lambda_2 \cdots \lambda_n|} \right)^{\frac{1}{2}} , \quad D_g = \left(\frac{|\lambda_1 \lambda_2 \cdots \lambda_n|}{{\lambda_1}^n} \right)^{\frac{1}{2}} . \tag{2.6}$$

In the two-dimensional case, Definition 2.2 and Definition 2.4 are equivalent, and provide two different approaches for analysis and computational convenience.

Lemma 2.8: If g is an affine transformation defined by Equation 2.1, then $D_g = \frac{\lambda_2}{\lambda_1}$, where, $\lambda_2 \geq \lambda_1$, are the eigenvalues of $A^t A$.

Proof: The proof follows immediately from Lemmas 2.5-2.7 by noting that $g' = A$ for an affine transformation defined by Equation 2.1. □

2.2.3. Strain Tensor

The purpose of this section is to show that the dilatation can also be evaluated through the use of strain tensors. Tensor analysis is an ideal tool to study entities that are independent of the choice of reference frames. A tensor is an abstract quantity represented in a particular reference frame by a set of

functions. These functions obey certain laws of transformation from one coordinate system to another. We consider a general transformation of a pair of n-dimensional coordinate systems, X, and $\bar{X}$:

$$\bar{x}^i = f^i(x^1, x^2, \cdots, x^n), \quad \text{for } i=1,2,...,n ,$$

$$x^i = g^i(\bar{x}^1, \bar{x}^2, \cdots, \bar{x}^n), \quad \text{for } i=1,2,...,n ,$$

where f^i's and g^i's are functions that map X to $\bar{X}$ and $\bar{X}$ to X, respectively.

Definition 2.9: A quantity, represented by the components, $\bar{T}^{i_1 i_2 \cdots i_k}{}_{j_1 \cdots j_m}$, and $T^{r_1 r_2 \cdots r_k}{}_{t_1 \cdots t_m}$, in the $\bar{X}$ and X reference frame, respectively, is said to be a mixed tensor with k contravariant components, m covariant components, and of weight N, if the components obey the following rule of transformation:

$$\bar{T}^{i_1 i_2 \cdots i_k}{}_{j_1 \cdots j_m}$$

$$= \left|\frac{\partial x}{\partial \bar{x}}\right|^N \frac{\partial \bar{x}^{i_1}}{\partial x^{r_1}} \cdots \frac{\partial \bar{x}^{i_k}}{\partial x^{r_k}} \frac{\partial x^{t_1}}{\partial \bar{x}^{j_1}} \cdots \frac{\partial x^{t_m}}{\partial \bar{x}^{j_m}} T^{r_1 r_2 \cdots r_k}{}_{t_1 \cdots t_m}, \qquad (2.7)$$

where the general summation rule is assumed, that is, $x^i x^i = x^1 x^1 + x^2 x^2 + \cdots + x^n x^n$, and $\left|\frac{\partial x}{\partial \bar{x}}\right|$ is the Jacobian of the coordinate transformation.

The above is a generalized form of Brand's definition [Bra57]. In general, the component, $T^{r_1 r_2 \cdots r_k}{}_{t_1 \cdots t_m}$, is simply referred to as a *tensor*. If N=0, $T^{r_1 r_2 \cdots r_k}{}_{t_1 \cdots t_m}$ is known as an *absolute tensor*. If k=0, it is known as a *covariant tensor*. If m=0, it is then known as a *contravariant tensor*. This section is not intended to explore the details of tensor theory which can be found in many texts [Sok51, Bra57]. A strain tensor provides another approach for computing the dilatation. Let $g:\Omega \rightarrow \bar{\Omega}$, $\Omega, \bar{\Omega} \subset \mathbb{R}^n$, be a homeomorphic mapping with orthogonal cartesian reference frame X, and $\bar{X}$, respectively. A *homeomorphic mapping* is a one-to-one mapping such that both the mapping and its inverse are continuous [Gug63]. The square of the arc length of Ω and $\bar{\Omega}$, using the general summation rule, are, respectively:

$$ds^2 = dx^i dx^i \quad \text{and} \quad d\bar{s}^2 = d\bar{x}^i d\bar{x}^i \quad , \quad i=1,2, \cdots ,n.$$

Thus, $d\bar{s}^2 - ds^2 = d\bar{x}^k d\bar{x}^k - dx^k dx^k$

$$= \frac{\partial \bar{x}^k}{\partial x^i}\frac{\partial \bar{x}^k}{\partial x^j} dx^i dx^j - \delta_{ij} dx^i dx^j$$

$$= 2\mu_{ij} dx^i dx^j \quad , \qquad i,j,k=1,2, \cdots ,n,$$

where $\mu_{ij} = \frac{1}{2}\left\{ \frac{\partial \bar{x}^k}{\partial x^i}\frac{\partial \bar{x}^k}{\partial x^j} - \delta_{ij} \right\}$.

Therefore, $\frac{ds^2 - d\bar{s}^2}{2ds^2} = \mu_{ij}\frac{dx^i}{ds}\frac{dx^j}{ds}$. (2.8)

Definition 2.10: The tensor μ_{ij}, which is symmetric ($\mu_{ij} = \mu_{ji}$), is known as the strain tensor [Sok51].

Note that the strain tensor is related to $g'^t g'$ of the quasiconformal mapping (Definition 2.4) as follows:

$$2\mu_{ij} = g'^t g' - I \quad ,$$

where I is the identity matrix.

The eigenvalues of μ_{ij} are known as the principal strains, and the corresponding eigenvectors are orthogonal. The directions corresponding to the principal strains are known as the principal directions (axes) of the strain tensor. At any point, it is a mapping from a ball to an ellipsoid. The ratio of the length of each semi-axis of the ellipsoid to the length of the radius of the ball is

$$\sqrt{1+2\lambda_i} \ , \ i=1,2,...,n \ , \tag{2.9}$$

where λ_i's are the eigenvalues of μ_{ij}.

The inner, outer, and maximal dilatations can also be obtained from the above relationship (Equation 2.9). For example, let the homeomorphic mapping be the affine transformation defined by Equation 2.1. The symmetric tensor, μ_{ij}, $i,j = 1,2$, for the affine transformation is

$$\frac{1}{2}\begin{bmatrix} a^2 + c^2 - 1 & ab + cd \\ ab + cd & b^2 + d^2 - 1 \end{bmatrix}. \tag{2.10}$$

The dilatation of the affine transformation can thus be evaluated by finding the eigenvalues of Equation 2.10 and using the relationship given by Equation 2.9.

2.3. Invariant Properties of Dilatation

In this section, the invariant properties of the dilatation will be discussed.

Lemma 2.11: The dilatation of a triangular transformation is translation invariant.

Proof: From Equation 2.4, the dilatation is not a function of the translational part of the affine transformation, and thus invariant to translation. □

Lemma 2.12: The dilatation of a triangular transformation is rotation invariant.

Proof: From Equation 2.4, it is sufficient to show that the terms, $(a+d)^2+(c-b)^2$, and $(a-d)^2+(c+b)^2$, are invariant to rotation. When the image points are rotated, that is,

$$\begin{bmatrix}\tilde{u}\\ \tilde{v}\end{bmatrix} = \begin{bmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}u\\ v\end{bmatrix}$$

$$= \begin{bmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}a & b\\ c & d\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix} + \begin{bmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}e\\ f\end{bmatrix}$$

$$= \begin{bmatrix}\tilde{a} & \tilde{b}\\ \tilde{c} & \tilde{d}\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix} + \begin{bmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}e\\ f\end{bmatrix} ,$$

it can be seen that

$$(\tilde{a}+\tilde{d})^2+(\tilde{c}-\tilde{b})^2 = (a+d)^2+(c-b)^2,$$

and

$$(\tilde{a}-\tilde{d})^2+(\tilde{c}+\tilde{b})^2 = (a-d)^2+(c+b)^2 .$$ □

Lemma 2.13: The dilatation of a triangular transformation is scale invariant.

Proof: Scaling of image points corresponds to multiplying the coefficients of the affine transformation by a constant. From Equation 2.4, the dilatation is thus invariant to scaling. □

The following theorem is thus obtained from the above three lemmas:

Theorem 2.14: The dilatation of a triangular transformation is invariant under the group of transformations $\mathbf{G} = \{$ translation, rotation, scaling$\}$.

Proof: This follows immediately by Lemmas 2.11-2.13. □

The above invariant properties are not restricted only to the triangular transformation. They also hold for a diffeomorphism.

Theorem 2.15: The dilatations of Definition 2.4 for a quasiconformal mapping are invariant under the group of transformations $\mathbf{G} = \{$ translation, rotation, scaling$\}$.

Proof: Let g be a diffeomorphism, and assume that g is mapped to $\bar{g}$ by $\mathbf{G}$, that is,

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

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

Thus,

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

$$= \alpha^2 g'^t (H^t H) g' = \alpha^2 g'^t g' \ .$$

From Lemmas 2.5 - 2.7, the desired invariance is obtained. □

Three definitions (Definitions 2.16, 2.17, and 2.18) that can be found in [Mui82] are given below in order to show that the dilatation of a triangular transformation is a maximal invariant (defined below). Denote $\mathbf{G}$ as a group of transformations from a space μ into itself.

Definition 2.16: Let $\mathbf{u}_1$, $\mathbf{u}_2$ be in μ ($\mathbf{u}_1, \mathbf{u}_2 \in \mu$). $\mathbf{u}_1$ is equivalent to $\mathbf{u}_2$ under $\mathbf{G}$, written as $\mathbf{u}_1 \sim \mathbf{u}_2 (mod \mathbf{G})$, if there exists a $S \in \mathbf{G}$ such that $\mathbf{u}_2 = S\mathbf{u}_1$ (S maps $\mathbf{u}_1$ to $\mathbf{u}_2$).

Definition 2.17: A function $\phi(\mathbf{u})$ defined on μ is said to be invariant under **G** if

$$\phi(\varsigma\mathbf{u}) = \phi(\mathbf{u}) \quad \text{for all} \quad \mathbf{u} \in \mu \quad \text{and} \quad \varsigma \in \mathbf{G}.$$

Definition 2.18: A function $\phi(\mathbf{u})$ defined on μ is said to be a maximal invariant under **G** if it is invariant under **G** and if

$$\phi(\mathbf{u}_1) = \phi(\mathbf{u}_2) \quad \text{implies} \quad \mathbf{u}_1 \sim \mathbf{u}_2 (mod\, \mathbf{G}).$$

From the above definitions, we can prove the following theorem:

Theorem 2.19: Let **u** be a set of three two-dimensional points, that is, $\mathbf{u} \in \mu = \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2$. Denote $D(\mathbf{u})$ as the dilatation derived from the affine transformation of a set of three fixed points, **x**, to **u**. $D(\mathbf{u})$ is a maximal invariant under $\mathbf{G} = \{$ translation, rotation, scaling$\}$.

Proof: Note that we have used $D(\mathbf{u})$ to indicate that it is **u** which is transformed by **G**. The dilatation is invariant under **G** by Theorem 2.14. To show that the dilatation is a maximal invariant under **G**, it is sufficient to show that if $\mathbf{u}_1 \in \mu$, and $\mathbf{u}_2 \in \mu$, then

$$D(\mathbf{u}_1) = D(\mathbf{u}_2) \quad \text{implies} \quad \mathbf{u}_2 = \varsigma\mathbf{u}_1 \text{ for some } \varsigma \in \mathbf{G}.$$

Given $\mathbf{u}_1$ and $\mathbf{u}_2$ in μ, there exist affine transformations that map the set of three fixed points **x** to $\mathbf{u}_1$ and $\mathbf{u}_2$, respectively. Let A_1 and $\mathbf{t}_1$ be the linear part and the translation part of the affine transformation that maps **x** to $\mathbf{u}_1$, and likewise for A_2 and $\mathbf{t}_2$.
From Lemma 2.8, if $D(\mathbf{u}_1) = D(\mathbf{u}_2)$, 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 vertical 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 dilatation.

Therefore, $\mathbf{u}_2 = \varsigma\mathbf{u}_1$. □

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

Theorem 2.20: If $\phi(\mathbf{u})$ on μ is a maximal invariant under **G**, then $\psi(\mathbf{u})$ is invariant under **G** if and only if ψ is a function of $\phi(\mathbf{u})$.

Proof: The proof can be found in [Mui82]. □

2.4. Sphericity

Sphericity is an alternative shape measure relative to the dilatation mentioned earlier. This shape measure will be used throughout this report due to its properties. The sphericity has the same invariant properties of the dilatation. For a triangular transformation, as shown in Figure 2.1, 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 $= \frac{2\sqrt{d_1 d_2}}{d_1+d_2}$. We define the sphericity of a diffeomorphism as follows:

Definition 2.21: The sphericity of a diffeomorphism, $g:\Omega \rightarrow \bar{\Omega}$, $(\Omega, \bar{\Omega} \subset \mathbb{R}^n)$, for $\mathbf{x} \in \Omega$, is defined as

$$\Upsilon_g(\mathbf{x}) = \frac{\left(\det(g'^t g')\right)^{\frac{1}{n}}}{\left(\frac{1}{n} tr(g'^t g')\right)} \quad , \tag{2.11}$$

where $det()$ and $tr()$ are the determinant and the trace of a matrix.

For the notational convenience, the argument of the sphericity may be omitted.

Lemma 2.22: The sphericity can be expressed in terms of the eigenvalues of $g'^t g'$ as follows:

$$\Upsilon_g = \frac{\left(\lambda_1 \lambda_2 \cdots \lambda_n\right)^{\frac{1}{n}}}{\left(\frac{1}{n}(\lambda_1 + \lambda_2 + \cdots + \lambda_n)\right)} \quad , \tag{2.12}$$

where λ_i, i=1,2, $\cdots$,n , are the eigenvalues of $g'^t g'$.

Proof: Note that

$$\begin{aligned}\det(g'^t g') &= \det(\Phi^t D \Phi) \qquad \text{(spectral decomposition)} \\ &= \det(D \Phi^t \Phi) \\ &= \det(D) \\ &= \lambda_1 \lambda_2 \cdots \lambda_n ,\end{aligned}$$

and

$$\begin{aligned}tr(g'^t g') &= tr(\Phi^t D \Phi) \\ &= tr(D \Phi^t \Phi) \\ &= tr(D) \\ &= \lambda_1 + \lambda_2 + \cdots + \lambda_n .\end{aligned}$$

The result thus follows. □

Note that the sphericity is the ratio of the geometric mean to the arithmetic mean of the eigenvalues of $g'^t g'$. 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 [Bar76], the ratio of the geometric mean to the arithmetic mean is between 0 and 1. The sphericity thus takes on values in [0, 1]. If $\Upsilon_g(\mathbf{x}) = 1$ for all $\mathbf{x} \in \Omega$, $g(\mathbf{x})$ is a conformal mapping. For a triangular transformation, the sphericity is 1 if the two triangles are similar. The less similar the two triangles are, the smaller the value of the sphericity. We shall next derive the relationship between the sphericity and the dilatation, and the invariant properties of the sphericity.

Lemma 2.23: In the two-dimensional case, $\Upsilon_g = \dfrac{2\sqrt{D_g}}{1 + D_g}$, where D_g is the dilatation as defined in Definition 2.4.

Proof: The result is obtained by using Definition 2.4 and Lemma 2.22. □

Theorem 2.24: The sphericity of a triangular transformation is invariant under $\mathbf{G} = \{$ translation, rotation, scaling$\}$.

Proof: By using Theorems 2.14, 2.19, 2.20 and Lemma 2.23, the sphericity of a triangular transformation is invariant under **G**. □

Theorem 2.25: The sphericity of a diffeomorphism of Definition 2.21 is invariant when the diffeomorphism undergoes the group of transformations **G** = { translation, rotation, scaling}.

Proof: The proof is similar to Theorem 2.15. □

Theorem 2.26: The sphericity of a triangular transformation is a maximal invariant under **G** = { translation, rotation, scaling}.

Proof: The proof is similar to the proof of Theorem 2.19. □

Lemma 2.27 The sphericity of an affine transformation defined by Equation 2.1, for the case when $|A| > 0$, is:

$$\Upsilon_g = \frac{t_1^2 + t_3^2 - (t_2^2 + t_4^2)}{t_1^2 + t_2^2 + t_3^2 + t_4^2} \tag{2.13}$$

where $t_1 = a + d$,

$t_2 = a - d$,

$t_3 = b - c$, and

$t_4 = b + c$.

Note that Equation 2.13 expresses the sphericity, Υ_g , in terms of the coefficients of the affine transformation. It is equivalent to the ratio of the geometric mean to the arithmetic mean of the lengths of the principal axes of the inscribed ellipse, as shown in Figure 2.1.

Proof: From Equations 2.1 and 2.11,

$$\Upsilon_g = \left(\frac{\det(A^t A)}{(\frac{1}{2} tr(A^t A))^2} \right)^{\frac{1}{2}}$$

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

$$= \frac{2(ad - bc)}{(a^2 + b^2 + c^2 + d^2)}$$

$$= \frac{t_1^2 + t_3^2 - (t_2^2 + t_4^2)}{t_1^2 + t_2^2 + t_3^2 + t_4^2} . \qquad \square$$

Lemma 2.27 will be used to determine the probability density function of the sphericity of a triangular transformation in the following section.

2.5. The Probability Density Function of The Sphericity

As mentioned earlier, the sphericity is introduced because it has the same invariant properties of the dilatation and it is more mathematically tractable for analysis. The sphericity of a triangular transformation which maps three model landmarks to three scene landmarks is a local shape measure that indicates the similarity between the two sets of landmarks. 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 let $\{(x_1, y_1), (x_2, y_2), (x_3, y_3)\}$ be those of 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 $\{(x_1, y_1), (x_2, y_2), (x_3, y_3)\}$ match the scene landmarks $\{(u_1, v_1), (u_2, v_2), (u_3, v_3)\}$.

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 2.4 that the sphericity is rotation, translation, and scale invariant. In this section, we shall show that the sphericity is relatively robust with respect to distortion in the 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 1. We assume that the distortion in the scene landmarks can be modeled as "noise" added to the locations of the model landmarks by the following:

$$u_i = x_i + n_i \ , \qquad i = 1,2,3 \ , \quad \text{and}$$

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

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,

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

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

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

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

$u_1 \sim \mathbf{n}(x_1, \sigma^2)$ means that the random variable u_1 has a normal probability density function with mean x_1 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 thus indicates that the sphericity is robust with respect to the above model of distortion. We thus have:

Lemma 2.28: The vector $\underline{u}$ is a multivariate normal random vector with mean vector, $\underline{\mu}$, and covariance matrix, $\sigma^2 I$; that is, $\underline{u} \sim \mathbf{mvn}\ (\underline{\mu}, \sigma^2 I)$, where:

$$\underline{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix}, \quad \text{and} \quad \underline{\mu} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix}.$$

$\underline{u} \sim \mathbf{mvn}\ (\underline{\mu}, \sigma^2 I)$ means that the random vector $\underline{u}$ has a multivariate normal probability density function with mean vector $\underline{\mu}$ and covariance matrix $\sigma^2 I$.

Proof: The lemma follows from the above assumption. □

From Equation 2.13, the sphericity of a triangular transformation is described in terms of the coefficients of the linear part of the affine transformation. We shall next determine the probability density function of these coefficients with the above assumption.

Lemma 2.29 Let $\underline{a} = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$. The elements of $\underline{a}$ are the coefficients of the linear part of the affine transformation defined by Equation 2.1. Then, $\underline{a} \sim \mathbf{mvn}(\underline{\nu}, \Sigma)$, where

$$\underline{\nu} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & 0 & 0 \\ \sigma_{12} & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \sigma_1^2 & \sigma_{12} \\ 0 & 0 & \sigma_{12} & \sigma_2^2 \end{bmatrix},$$

and where

$$\sigma_1^2 = \left(\frac{\sigma}{2\Delta}\right)^2((y_2-y_3)^2+(y_3-y_1)^2+(y_1-y_2)^2),$$

$$\sigma_{12} = \left(\frac{\sigma}{2\Delta}\right)^2((y_2-y_3)(x_3-x_2)+(y_3-y_1)(x_1-x_3)+(y_1-y_2)(x_2-x_1)),$$

$$\sigma_2^2 = \left(\frac{\sigma}{2\Delta}\right)^2((x_3-x_2)^2+(x_1-x_3)^2+(x_2-x_1)^2), \text{ and}$$

$$\Delta = \frac{1}{2}\det\left(\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix}\right) \quad ..$$

= area of a triangle with vertices (x_1, y_1), (x_2, y_2), and (x_3, y_3).

Proof: From Equation 2.2, we have

$$\underline{a} = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = T\underline{u} \tag{2.14}$$

$$= \frac{1}{2\Delta}\begin{bmatrix} y_2-y_3 & y_3-y_1 & y_1-y_2 & 0 & 0 & 0 \\ x_3-x_2 & x_1-x_3 & x_2-x_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & y_2-y_3 & y_3-y_1 & y_1-y_2 \\ 0 & 0 & 0 & x_3-x_2 & x_1-x_3 & x_2-x_1 \end{bmatrix}\underline{u} \ .$$

If T has full rank, i.e., rank (T) = 4, $\underline{a}$ is multivariate normal. By noting the fact that rank (T) = 2 if and only if

$$\begin{bmatrix} y_2-y_3 \\ y_3-y_1 \\ y_1-y_2 \end{bmatrix} = (\text{non-zero constant}) \times \begin{bmatrix} x_3-x_2 \\ x_1-x_3 \\ x_2-x_1 \end{bmatrix} ;$$

that is, (x_1, y_1), (x_2, y_2), and (x_3, y_3) are collinear, it follows that T has full rank if and only if (x_1, y_1), (x_2, y_2), and (x_3, y_3) are not collinear.

Hence, $\underline{a} \sim \mathbf{mvn}(T\underline{\mu}, \sigma^2 TIT^t)$.

The expressions for $\underline{\mu}$ and Σ are obtained by further simplification. □

Note that Σ is block diagonal, and $\{a, b\}$ are statistically independent of $\{c, d\}$. From the expression of the sphericity (see Lemma 2.27), we make the following transformation:

$$\underline{t} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{bmatrix} = C\underline{a} \ ,$$

$$\text{where } C = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}.$$

Lemma 2.30 $\underline{t} \sim \mathbf{mvn}(\underline{\omega}, \Xi)$, where

$$\underline{\omega} = \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \text{and}$$

$$\Xi = \begin{bmatrix} \sigma_1^2+\sigma_2^2 & \sigma_1^2-\sigma_2^2 & 0 & 2\sigma_{12} \\ \sigma_1^2-\sigma_2^2 & \sigma_1^2+\sigma_2^2 & 2\sigma_{12} & 0 \\ 0 & 2\sigma_{12} & \sigma_1^2+\sigma_2^2 & -(\sigma_1^2-\sigma_2^2) \\ 2\sigma_{12} & 0 & -(\sigma_1^2-\sigma_2^2) & \sigma_1^2+\sigma_2^2 \end{bmatrix}.$$

Proof: Since C is orthogonal, we have $\underline{t} \sim \mathbf{mvn}(C\underline{\mu}, C\Sigma C^t)$. By direct algebraic manipulation, we obtain the above expressions for $\underline{\omega}$ and Ξ. □

Note that t_1 and t_3 are statistically independent, and so are t_2 and t_4. From the above lemma and Lemma 2.27, it is difficult to find the probability density function of the sphericity in closed form because the elements of $\underline{t}$ are correlated. We shall examine the special case where the elements of $\underline{t}$ are mutually independent. For this case, the probability density function of the sphericity will be shown to have a non-central Beta probability density function. The special case is the following lemma.

Lemma 2.31 If the set of points (x_1, y_1), (x_2, y_2), and (x_3, y_3) form an equilateral triangle, then $\underline{t} \sim \mathbf{mvn}(\begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \end{bmatrix}, (\sigma_1^2 + \sigma_2^2)I)$.

Proof: By simple geometric and algebraic manipulation, it is seen that if (x_1, y_1), (x_2, y_2), and (x_3, y_3) form an equilateral triangle,

$$(x_3-x_2)^2+(x_1-x_3)^2+(x_2-x_1)^2 = (y_2-y_3)^2+(y_3-y_1)^2+(y_1-y_2)^2$$

and $$(y_2-y_3)(x_3-x_2)+(y_3-y_1)(x_1-x_3)+(y_1-y_2)(x_2-x_1) = 0 \ .$$

Equivalently, $\sigma_1^2 - \sigma_2^2 = 0$ and $\sigma_{12} = 0$. Therefore, the elements of $\underline{t}$ are mutually independent. □

Before deriving the probability density function of the sphericity for this special case, we need to discuss the properties of the non-central Beta probability density function.

Definition 2.32 Let

$$U = \sum_{i=1}^{n_1} U_i^2 \ , \ \ U_i \sim \mathbf{n}(\alpha_i, \sigma^2) \ , \ \text{and}$$

$$V = \sum_{j=1}^{n_2} V_i^2 \ , \ \ V_i \sim \mathbf{n}(0, \sigma^2) \ ,$$

where U_i's and V_i's are statistically independent.

Then, the random variable $W = \dfrac{U}{U+V}$ has a non-central Beta probability density function [Hod55, Seb63] with n_1 and n_2 degrees of freedom, and the non-centrality, ρ, denoted by:

$$f_W(w) = \beta(w; \frac{n_1}{2}, \frac{n_2}{2}, \rho) \ ,$$

where $f_W(.)$ denotes the probability density function of the random variable W,

w is the variable of the function $f_W(.)$, and

$$\rho = \frac{1}{2\sigma^2} \sum_{i=1}^{n_1} \alpha_i{}^2 .$$

Note that W is (central) Beta distributed if $\rho = 0$. We shall use the notation of the probability density function of a non-central Beta random variable without the non-centrality term to denote the probability density function of a (central) Beta random variable.

Theorem 2.33 The probability density function of the random variable W described by Definition 2.32 can be expressed as an infinite sum of the

probability density functions of the central Beta random variables as follows:

$$f_W(w)=\beta(w;\frac{n_1}{2},\frac{n_2}{2},\rho) = \sum_{k=0}^{\infty} e^{-\rho}\frac{\rho^k}{k!}\beta(w;\frac{n_1}{2}+k,\frac{n_2}{2}),$$

where

$$\beta(w;\frac{n_1}{2}+k,\frac{n_2}{2}) = \frac{\Gamma(\frac{n_1}{2}+k+\frac{n_2}{2})}{\Gamma(\frac{n_1}{2}+k)\Gamma(\frac{n_2}{2})} w^{(\frac{n_1}{2}+k-1)}(1-w)^{(\frac{n_2}{2}-1)}$$

denotes the probability density function of a central Beta random variable with parameters $\frac{n_1}{2}+k$, $\frac{n_2}{2}$, and

Γ is the Γ-function [Kap81].

Proof: The proof can be found in [Gra61]. □

Having defined the non-central Beta probability density function, we can now derive the probability density function of the sphericity in closed form for the above special case.

Theorem 2.34 The sphericity derived from the triangular transformation of an equilateral triangle has the following probability density function:

$$f_{\Upsilon_g}(\upsilon_g) = \frac{1}{2}\beta(\frac{\upsilon_g+1}{2};\, 1,\, 1,\, \frac{2}{\sigma_1^2+\sigma_2^2}),$$

where $f_{\Upsilon_g}(.)$ is the probability density function of the sphericity, Υ_g, and υ_g is the variable of the function $f_{\Upsilon_g}(.)$.

Proof: Manipulating Equation 2.13, we have

$$\Upsilon_g = 2\frac{\frac{t_1^2+t_3^2}{t_2^2+t_4^2}}{\frac{t_1^2+t_3^2}{t_2^2+t_4^2}+1} - 1.$$

Note that t_1, t_2, t_3, and t_4 are mutually independent. By Definition 2.32, the random variable $\frac{\frac{t_1^2+t_3^2}{t_2^2+t_4^2}}{\frac{t_1^2+t_3^2}{t_2^2+t_4^2}+1}$ has a non-central Beta

probability density function with 2, 2 degrees of freedom, and noncentrality, $\frac{2}{\sigma_1^2+\sigma_2^2}$. By simple transformation, the probability density function of Υ_g is obtained. □

We shall later show that $\frac{t_1^2 + t_3^2}{t_2^2 + t_4^2}$ ($g_D = \frac{t_1^2 + t_3^2}{t_2^2 + t_4^2}$, see Equation 2.4) is a non-central F-distributed random variable [Joh70]. We shall next compute the mean and variance of the sphericity. Hence, we must compute the mean and the variance of a non-central Beta random variable. We shall briefly show how to compute the mean and variance using Theorem 2.33. A detailed proof of Lemma 2.36 is provided in the Appendix.

Lemma 2.35 Let W be a Beta random variable with parameters p,q. That is, $f_W(w) = \beta(w;p,q)$. Then,

$$E(W) = \frac{p}{p+q},$$

$$E(W^2) = \left(\frac{p}{p+q}\right)\left(\frac{p+1}{p+q+1}\right),$$

$$Var(W) = \frac{pq}{(p+q)^2(p+q+1)}.$$

Proof: This is verified by noting the following [Bic77]:

$$\int_0^1 w^{p-1}(1-w)^{q-1}dw = \frac{\Gamma(p)\Gamma(q)}{\Gamma(p+q)}.$$ □

Lemma 2.36 If W is a non-central Beta random variable, with 2, 2 degrees of freedom, and the noncentrality, ρ, that is, $f_W(w) = \beta(w;1,1,\rho)$, then

$$E(W) = 1 - \frac{1}{\rho} + \frac{1}{\rho^2} - \frac{1}{\rho^2}e^{-\rho},$$

$$E(W^2) = 1 - \frac{2}{\rho} + \frac{4}{\rho^2}(1 - \frac{1}{\rho} + \frac{e^{-\rho}}{\rho}),$$

$$Var(W) = \frac{1}{\rho^2}(1 - \frac{2}{\rho} - \frac{1}{\rho^2} + 2e^{-\rho}(1 + \frac{1}{\rho} + \frac{1}{\rho^2} + \frac{e^{-\rho}}{\rho^2})) .$$

Proof: It is not obvious that there are any published results on the moments of a non-central Beta random variable. The moments can be computed by noting that the probability density function of a non-central Beta random variable can be expressed in terms of an infinite sum of the product of the probability density functions of the Poisson and the central Beta random variables in a separable fashion as indicated by Theorem 2.33. The moments of a Poisson random variable can be obtained by using its moment-generating function. The mean and the variance of a central Beta random variables are given in Lemma 2.35. With further algebraic manipulation, the desired result is obtained. A detailed proof of this lemma is provided in the Appendix. □

In general, the moments of a non-central Beta random variable with other degrees of freedom can be obtained in a similar way. We shall now determine the mean and the variance of the sphericity.

Theorem 2.37 Using the same notation and assumption of Theorem 2.34, we have

$$E(\Upsilon_g) = 1 - \frac{2}{\rho} + \frac{2}{\rho^2} - \frac{2}{\rho^2}e^{-\rho} , \tag{2.15}$$

$$Var(\Upsilon_g) = \frac{4}{\rho^2}(1 - \frac{2}{\rho} - \frac{1}{\rho^2} + 2e^{-\rho}(1 + \frac{1}{\rho} + \frac{1}{\rho^2} + \frac{e^{-\rho}}{\rho^2})) . \tag{2.16}$$

Proof: The result is obtained by using Lemma 2.36 and Theorem 2.34. □

For an equilateral triangle, the non-centrality of the sphericity, $\rho = \frac{(\text{sidelength})^2}{2\sigma^2}$. If the set of three scene landmarks are duplicates of the three model landmarks, the sphericity derived from mapping the model landmarks to the scene landmarks should be 1. With minor distortion in the landmark locations, we would hope that the sphericity would be close to 1; i.e., we would hope the sphericity has a mean close to 1 and variance close to 0. This indicates that the sphericity is relatively robust with respect to the distortion. Minor distortion in the landmark locations implies that the variance, σ^2, of the i.i.d. normal random variables corresponding to the distortion in the scene landmarks is small. From Equation 2.15 and 2.16, if

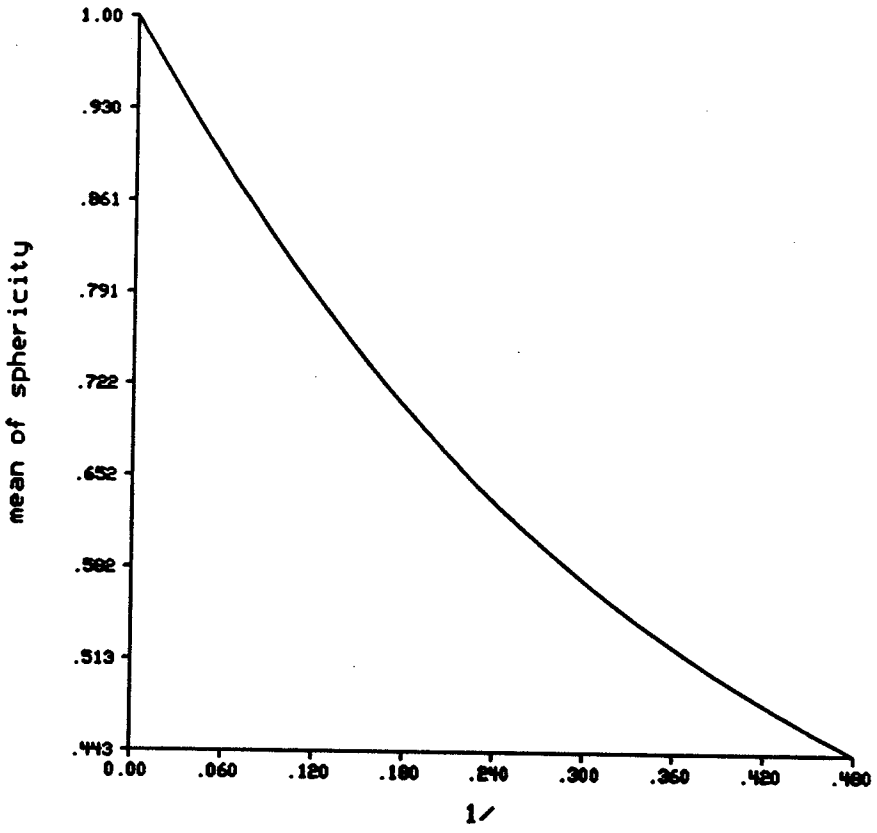


Figure 2.2. A plot of the mean of the sphericity, $E(\Upsilon_g)$, given by Equation 2.15 versus $\frac{1}{\rho}$. Note that $\frac{1}{\rho} = \frac{2\sigma^2}{(\text{sidelength})^2}$, where σ and sidelength are the standard deviation used for the i.i.d. zero mean normal random variables that model the distortion in the scene landmarks, and the length of a side of the equilateral triangle, respectively.

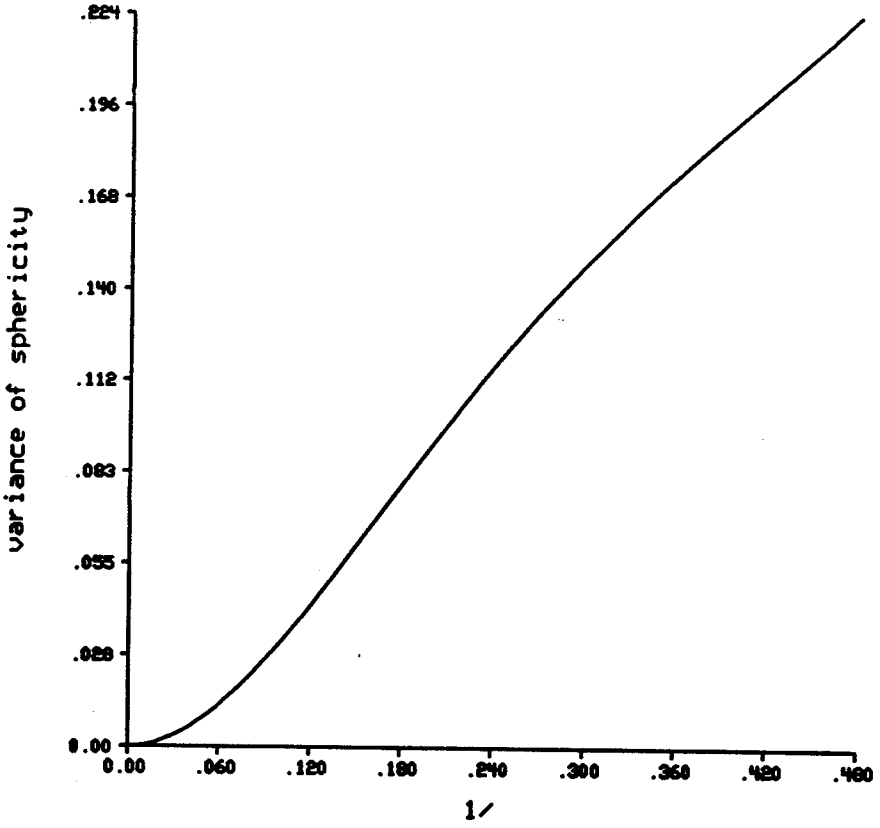


Figure 2.3. A plot of the variance of the sphericity, $Var(\Upsilon_g)$, given by Equation 2.16 versus $\frac{1}{\rho}$. Note that $\frac{1}{\rho} = \frac{2\sigma^2}{(\text{sidelength})^2}$, where σ and sidelength are the standard deviation used for the i.i.d. zero mean normal random variables that model the distortion in the scene landmarks, and the length of a side of the equilateral triangle, respectively.

$\sigma^2 \rightarrow 0$,

$$E(\Upsilon_g) \rightarrow 1 \quad \text{and} \quad Var(\Upsilon_g) \rightarrow 0 .$$

The mean (Equation 2.15) and the variance (Equation 2.16) of the sphericity are shown in Figures 2.2 and 2.3. Note that if $\frac{1}{\rho} < 0.02$ ($\frac{\sigma}{\text{sidelength}} < 0.1$), $Var(\Upsilon_g) < 0.015$ and $E(\Upsilon_g) > 0.96$. 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 are shown in Figures 2.2 and 2.3. Hence, the sphericity is relatively robust with respect to distortion in the sense that a small perturbation in the landmark locations does not significantly change its value.

We shall next define a non-central F random variable and show that the conformity (see Equation 2.4) has a non-central F probability density function.

Definition 2.38 Let

$$U = \sum_{i=1}^{n_1} U_i^2 , \quad U_i \sim N(\alpha_i, 1) , \text{ and}$$

$$V = \sum_{j=1}^{n_2} V_i^2 , \quad V_i \sim N(0, 1) ,$$

where U_i's and V_i's are statistically independent.

Then, the random variable $\frac{n_2}{n_1}\frac{U}{V}$ is non-central F-distributed [Joh70], with n_1, n_2 degrees of freedom, and non-centrality, $\sum_{i=1}^{n_1} \alpha_i^2$.

Theorem 2.39 The conformity derived from the triangular transformation of an equilateral triangle with the above assumptions is non-central F-distributed with 2, 2 degrees of freedom and non-centrality, $\frac{4}{\sigma_1^2 + \sigma_2^2}$.

Proof: From Equation 2.4,

$$g_D = \frac{t_1^2 + t_3^2}{t_2^2 + t_4^2}.$$

Dividing both the numerator and the denominator of g_D by $(\sigma_1^2 + \sigma_2^2)$, it is seen that the random variable g_D is non-central F-distributed by Lemma 2.31 and Definition 2.38. □

Since the mean and the variance of a non-central F random variable do not exist for 2, 2 degrees of freedom [Joh70], likewise, the mean and the variance of the conformity do not exist.

We have derived the probability density function, the mean, and the variance of the sphericity of a triangular transformation in closed form for the special case of an equilateral triangle. We shall next empirically estimate the probability density function of the sphericity for cases where the probability density function cannot be obtained in closed form.

2.6. Empirical Estimation of The Probability Density Function of Sphericity and Dilatation

We have derived the probability density function of the sphericity in closed form for the special case of the equilateral triangle. It is not clear how the sphericity is distributed for the general case. With the same assumptions used in the case of modeling the distortion in the scene landmarks described in the previous section, we empirically estimate the probability density function, the mean, and the standard deviation of the sphericity for several types of triangles using histograms. Each type of triangle 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 Figure 2.4. 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, as shown in Figure 2.5. 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 the estimation for each case.

The sphericity is distributed on [0, 1], which is quantized into 50 regions for the cases studied. Figures 2.6-2.11 show the estimated probability density function of the sphericity for six 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 in the scene landmarks. Figures 2.12 and 2.13 show some profiles of the plots shown in Figure 2.7 and 2.9, respectively. The estimated mean of the sphericity for various types of triangles and "noise levels" is shown in Figure 2.14, and the estimated standard deviation is shown in Figure 2.15. With a low noise level, the sphericity has an estimated mean close to 1, and standard deviation close to 0. The sphericity is thus relatively robust with respect to distortion in the sense that a small perturbation in the landmark locations does not significantly change its value.

With the same assumptions as mentioned above, we also estimate the probability density function, the mean, and the standard deviation of the dilatation using histograms as shown in Figures 2.16-25. From Equation 2.4, the dilatation is distributed on $[1.0, \infty)$. Since samples of the dilatation that fall beyond 5.0 are negligible for the cases studied, only the range [1.0, 5.0] is shown in Figures 2.14-23. This range is also quantized into 50 regions. With a low noise level, the dilatation has an estimated mean close to 1, and a very small standard deviation compared to the range of the dilatation. This indicates that the dilatation is also relatively robust with respect to distortion. Since the value of the dilatation may be too large for computer manipulation, the use of the sphericity as a local shape measure is preferable.

2.7. Summary

Two local shape measures, the dilatation and the sphericity, have been studied in detail. We have defined the dilatation of a triangular transformation, and shown that it can be evaluated by three different mathematical approaches: direct geometric method, quasiconformal mapping, and through the use of strain tensors. The sphericity has also been introduced and defined as a shape measure.

Both the dilatation and the sphericity of a triangular transformation are translation, rotation, and scale invariant. To demonstrate the effect of these shape measures with respect to distortion in the scene landmark locations, the coordinates of the model landmarks are assumed to be corrupted and modeled as random variables. For a set of three model landmarks that form an equilateral triangle, the probability density function of the sphericity of the triangular transformation is shown to have a non-central Beta probability density function. The probability density function of both the sphericity and

dilatation for cases which cannot be obtained in closed form are empirically estimated. We conclude from these results that these shape measures are relatively robust with respect to distortion. The following chapters will demonstrate the application of the sphericity for object recognition.

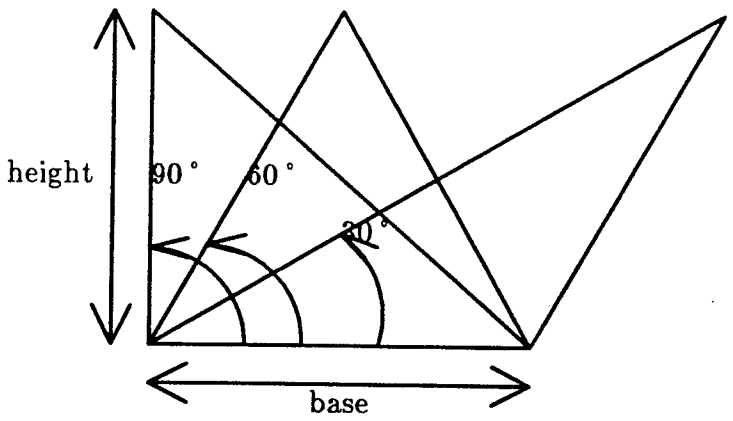


Figure 2.4. Types of triangles used for estimating probability density function of the sphericity and dilatation: each specified by an angle.

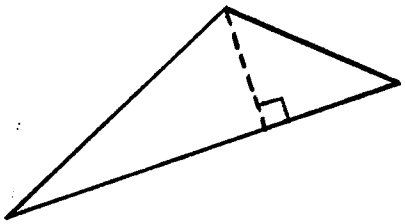


Figure 2.5. An example showing the smallest perpendicular distance from a vertex to the opposite side of a triangle.

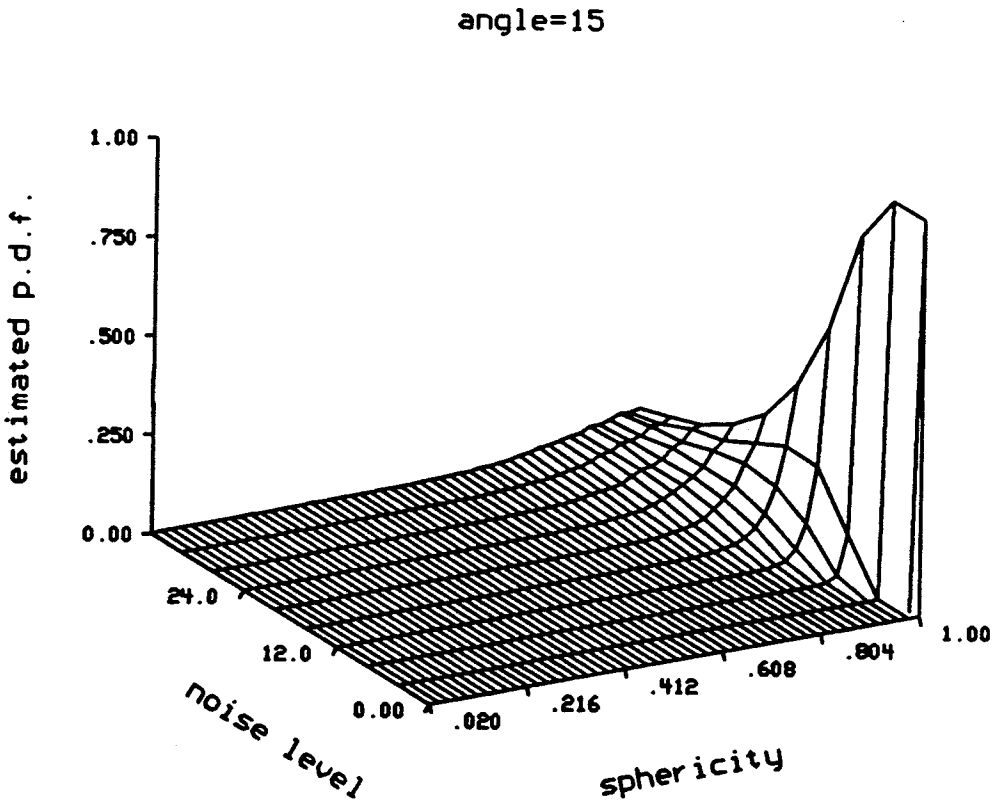


Figure 2.6. Estimated probability density function of the sphericity for angle=15° at various noise levels. Each value of the "noise level" corresponds to a percentage of the smallest perpendicular distance of the triangle. The length corresponding to the percentage value is used as the standard deviation of the zero mean i.i.d. normal random variables used for modeling the distortion in the landmarks.

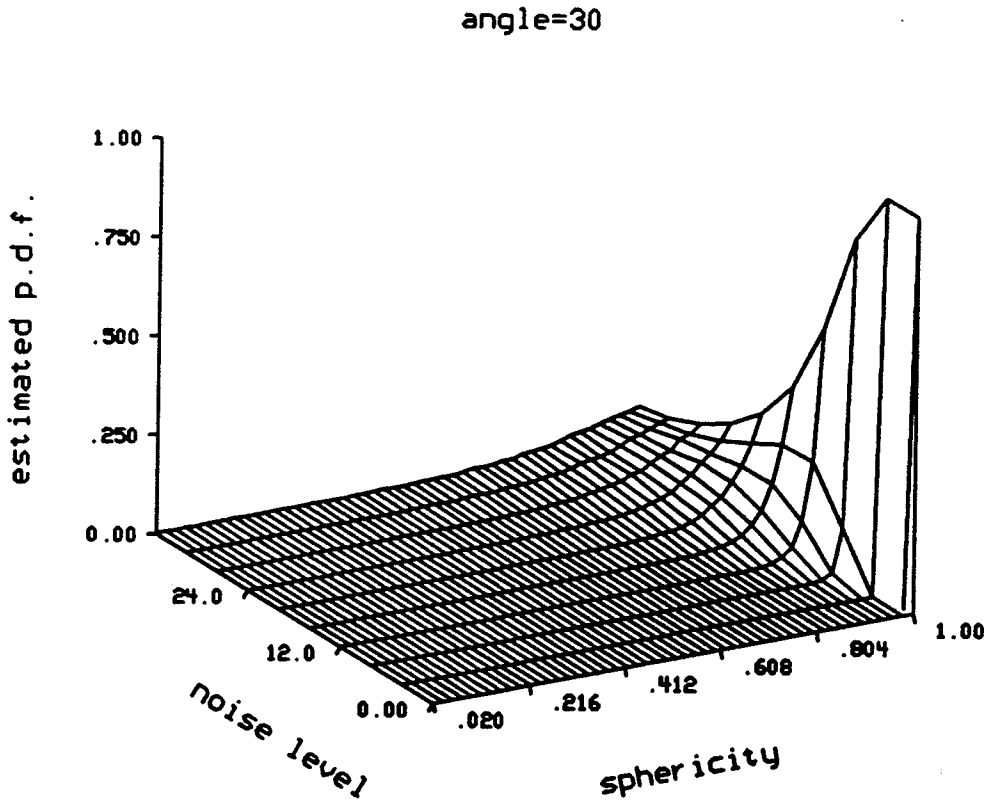


Figure 2.7. Estimated probability density function of the sphericity for angle=30° at various noise levels. Each value of the "noise level" corresponds to a percentage of the smallest perpendicular distance of the triangle. The length corresponding to the percentage value is used as the standard deviation of the zero mean i.i.d. normal random variables used for modeling the distortion in the landmarks.

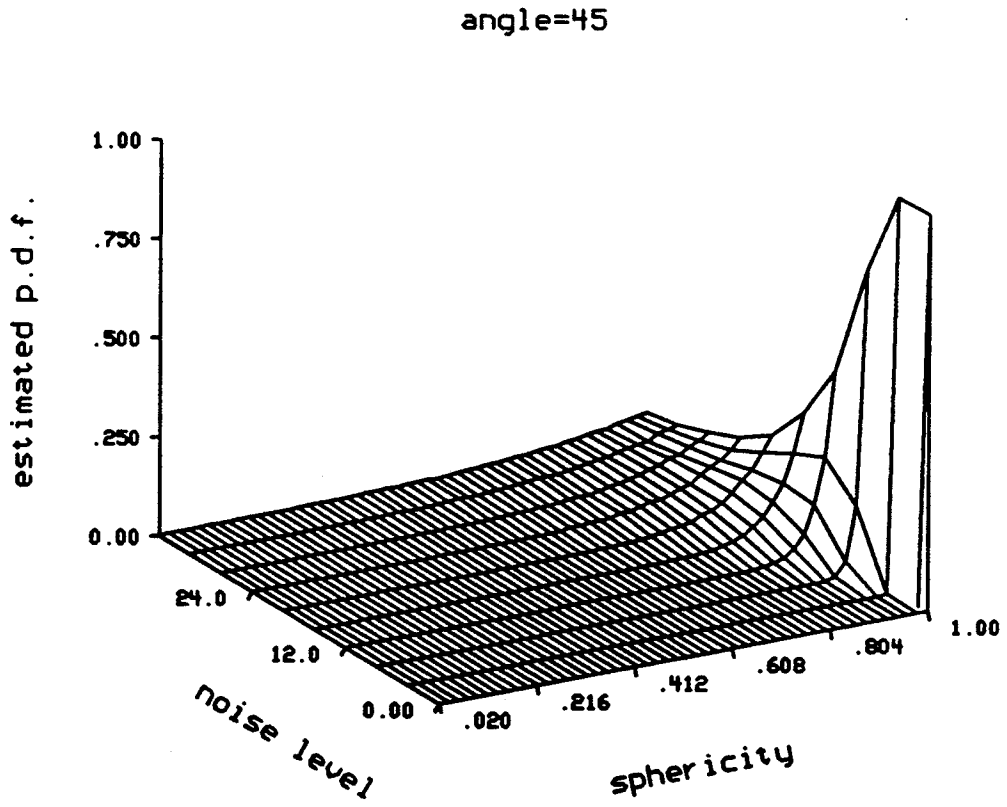


Figure 2.8. Estimated probability density function of the sphericity for angle=45° at various noise levels. Each value of the "noise level" corresponds to a percentage of the smallest perpendicular distance of the triangle. The length corresponding to the percentage value is used as the standard deviation of the zero mean i.i.d. normal random variables used for modeling the distortion in the landmarks.

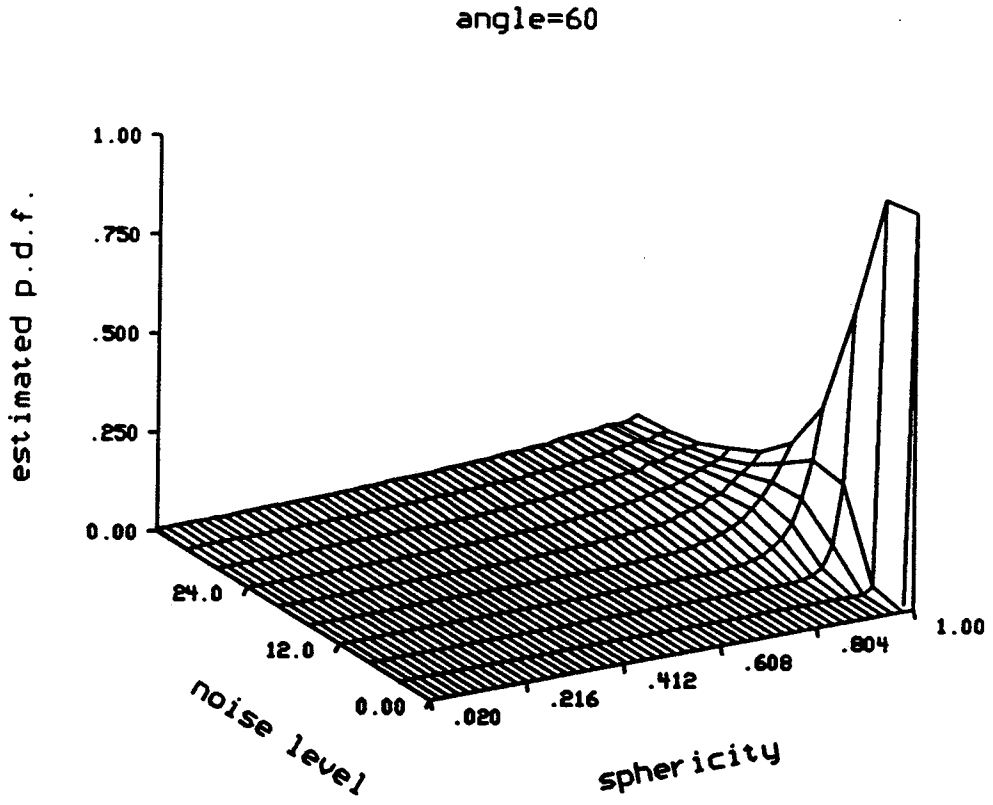


Figure 2.9. Estimated probability density function of the sphericity for angle=60° at various noise levels. Each value of the "noise level" corresponds to a percentage of the smallest perpendicular distance of the triangle. The length corresponding to the percentage value is used as the standard deviation of the zero mean i.i.d. normal random variables used for modeling the distortion in the landmarks.

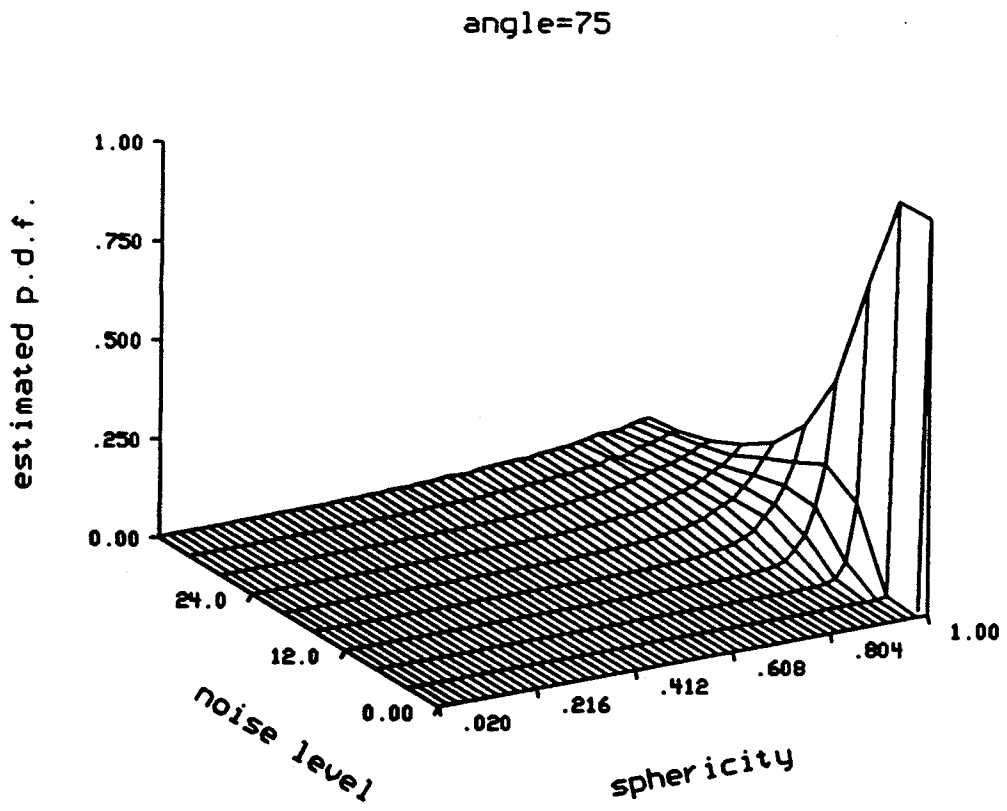


Figure 2.10. Estimated probability density function of the sphericity for angle=75° at various noise levels. Each value of the "noise level" corresponds to a percentage of the smallest perpendicular distance of the triangle. The length corresponding to the percentage value is used as the standard deviation of the zero mean i.i.d. normal random variables used for modeling the distortion in the landmarks.

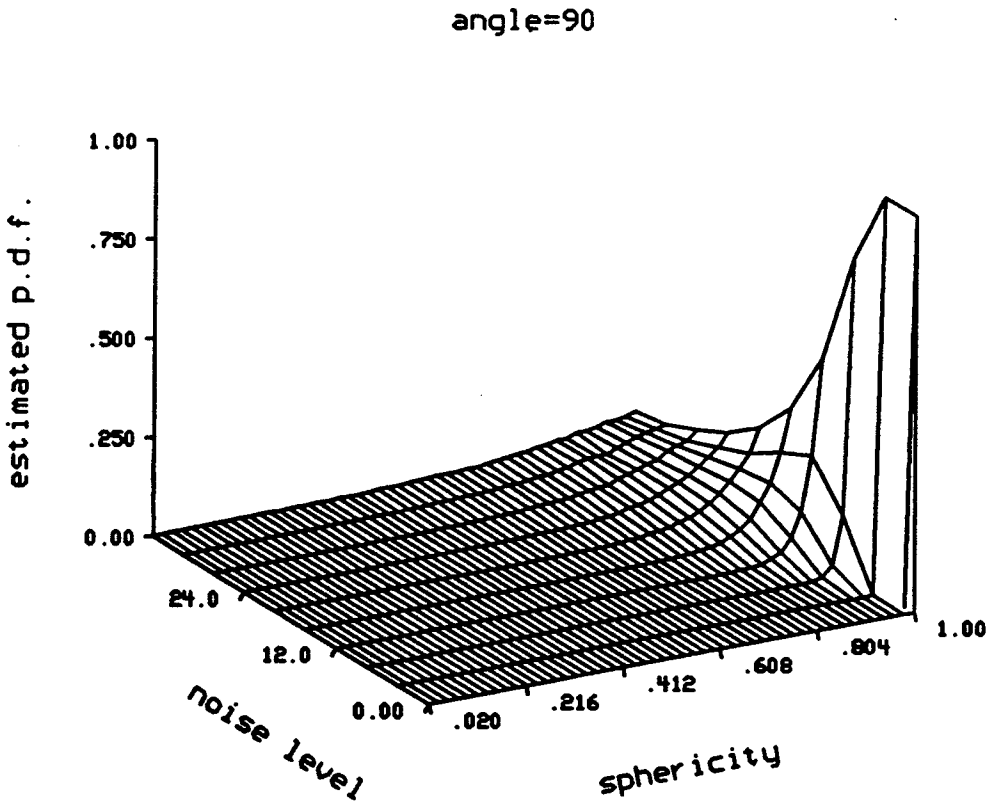


Figure 2.11. Estimated probability density function of the sphericity for angle=90° at various noise levels. Each value of the "noise level" corresponds to a percentage of the smallest perpendicular distance of the triangle. The length corresponding to the percentage value is used as the standard deviation of the zero mean i.i.d. normal random variables used for modeling the distortion in the landmarks.

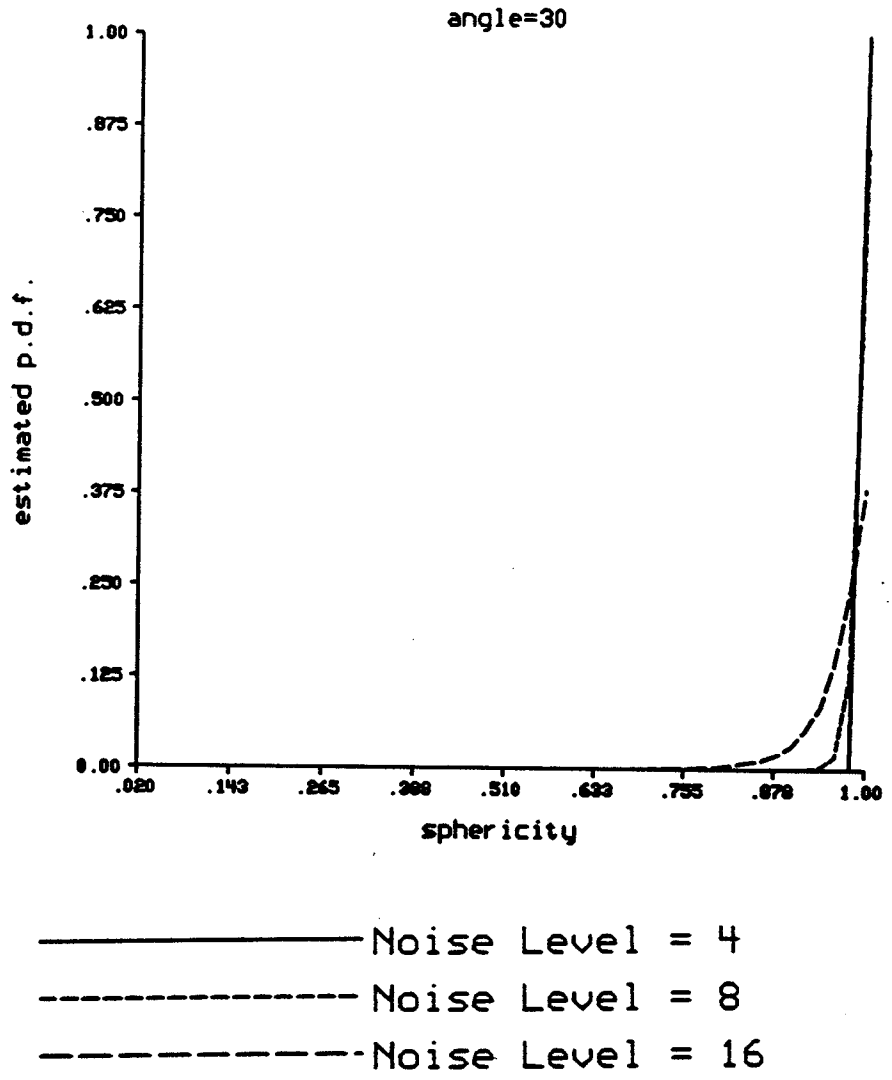


Figure 2.12. Profiles of the plot of Figure 2.7. They correspond to the estimated probability density function of the sphericity for angle=30° at noise levels of 4, 8, and 16.

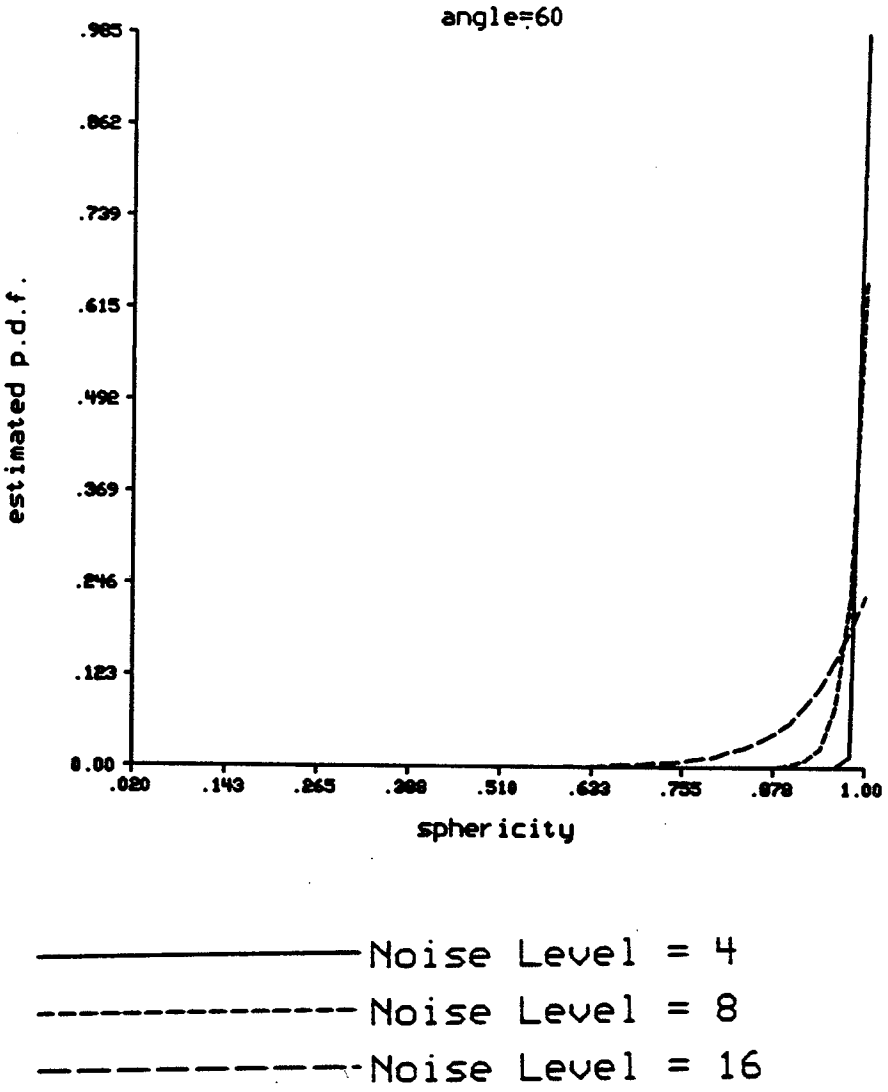


Figure 2.13. Profiles of the plot of Figure 2.9. They correspond to the estimated probability density function of the sphericity for angle=60 ° at noise levels of 4, 8, and 16.

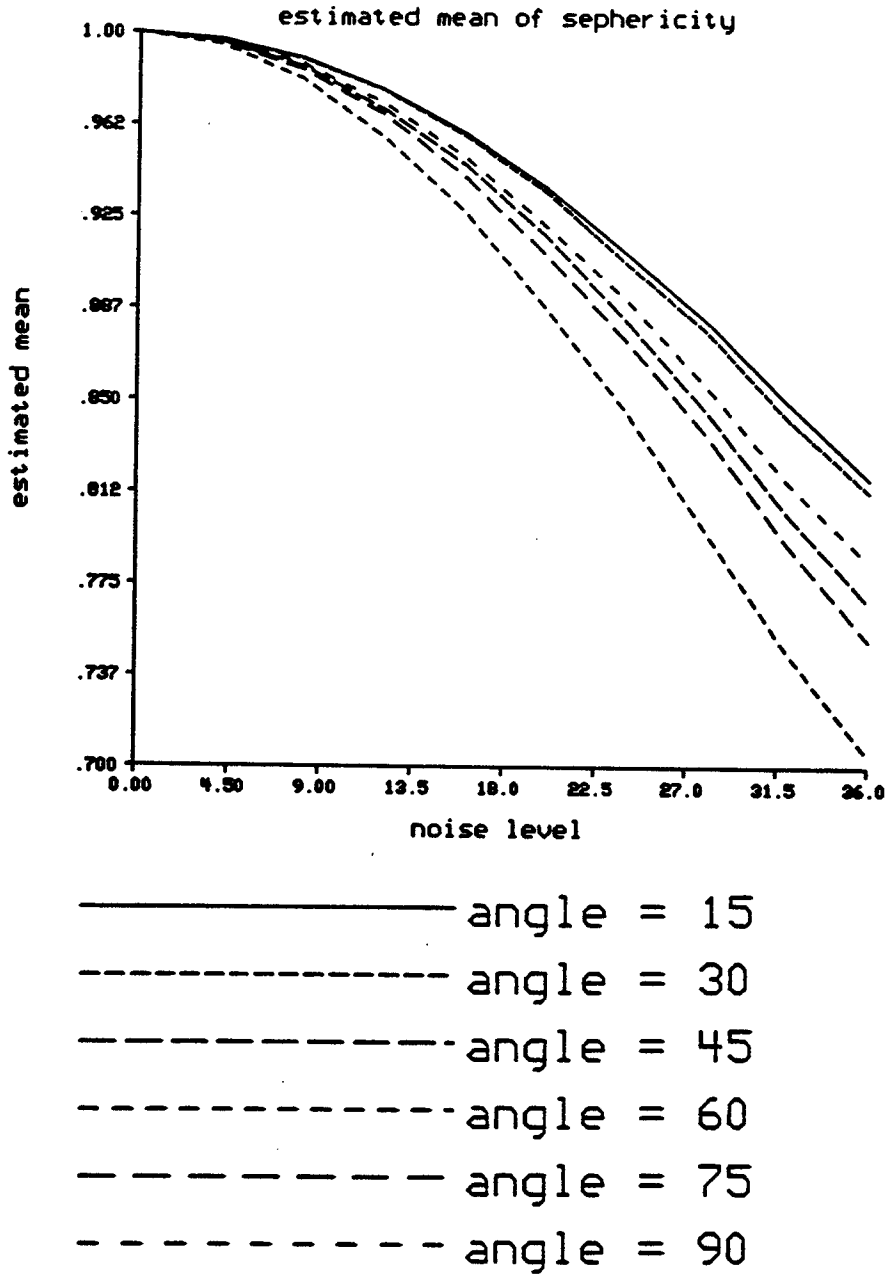


Figure 2.14. Estimated mean of the sphericity corresponding to different types of triangles specified by angles 15°, 30°, 45°, 60°, 75°, and 90°.

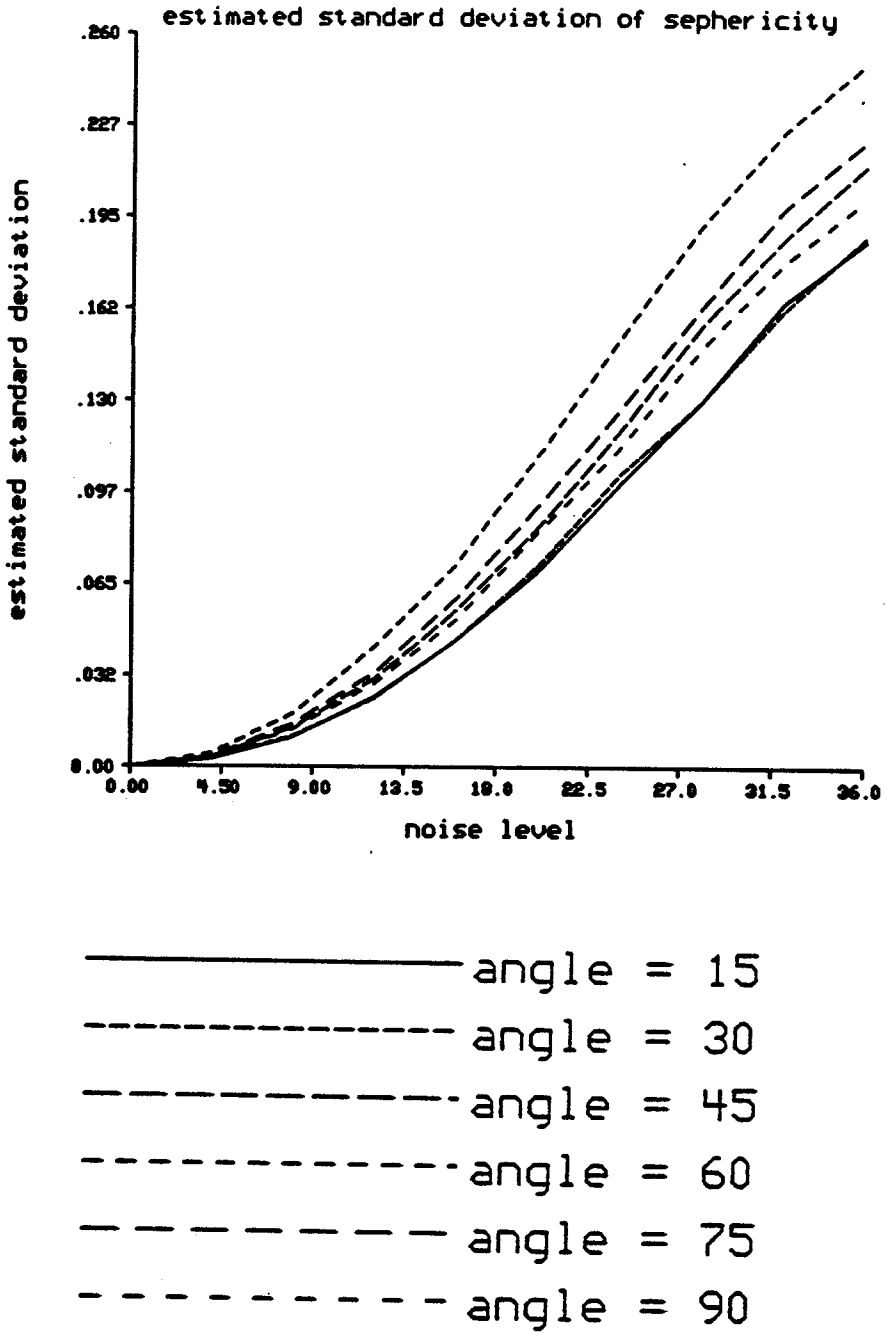


Figure 2.15. Estimated standard deviation of the sphericity corresponding to different types of triangles specified by angles 15°, 30°, 45°, 60°, 75°, and 90°.

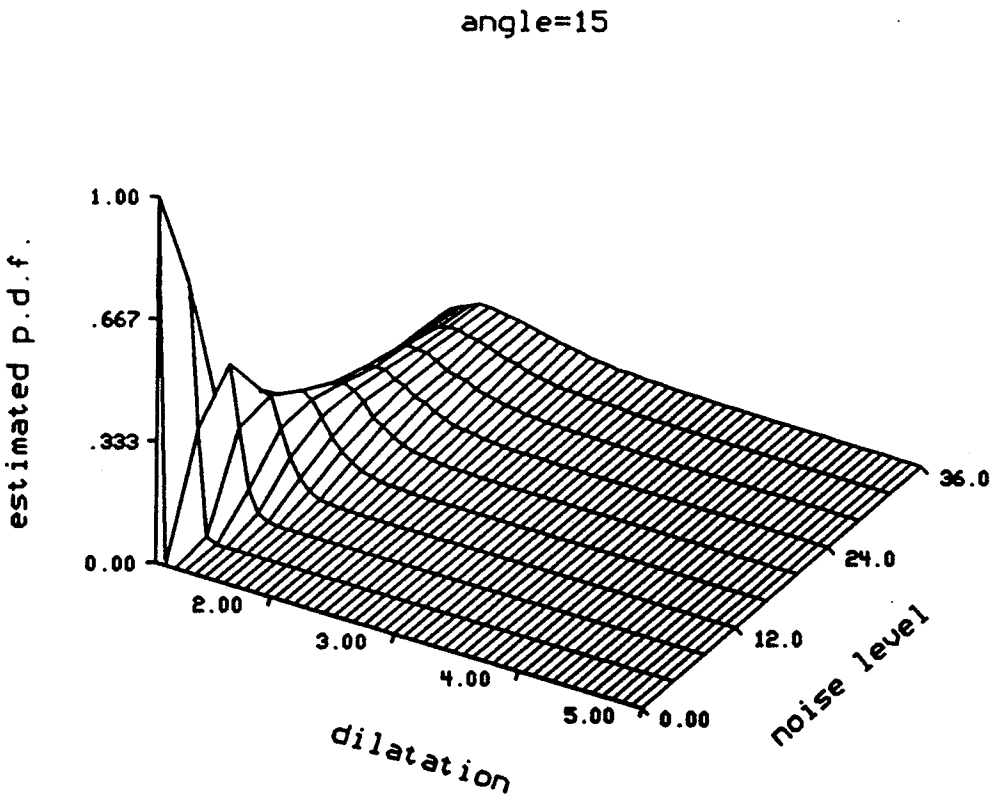


Figure 2.16. Estimated probability density function of the dilatation for angle=15° at various noise levels. Each value of the "noise level" corresponds to a percentage of the smallest perpendicular distance of the triangle. The length corresponding to the percentage value is used as the standard deviation of the zero mean i.i.d. normal random variables used for modeling the distortion in the landmarks.

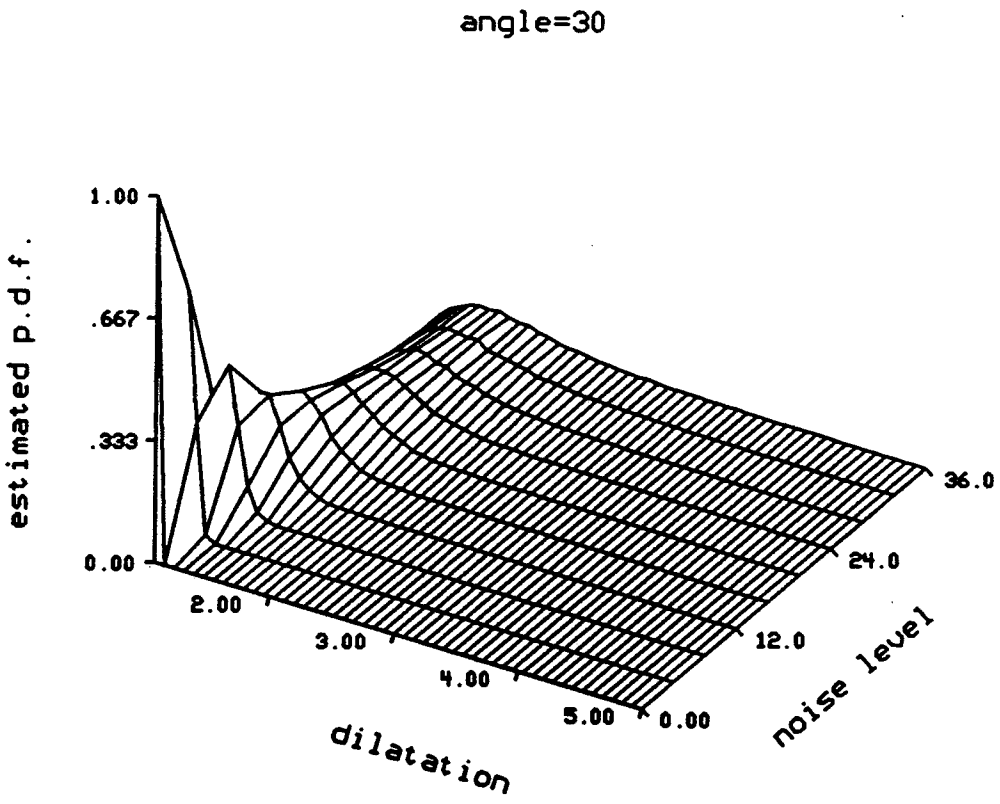


Figure 2.17. Estimated probability density function of the dilatation for angle=30° at various noise levels. Each value of the "noise level" corresponds to a percentage of the smallest perpendicular distance of the triangle. The length corresponding to the percentage value is used as the standard deviation of the zero mean i.i.d. normal random variables used for modeling the distortion in the landmarks.

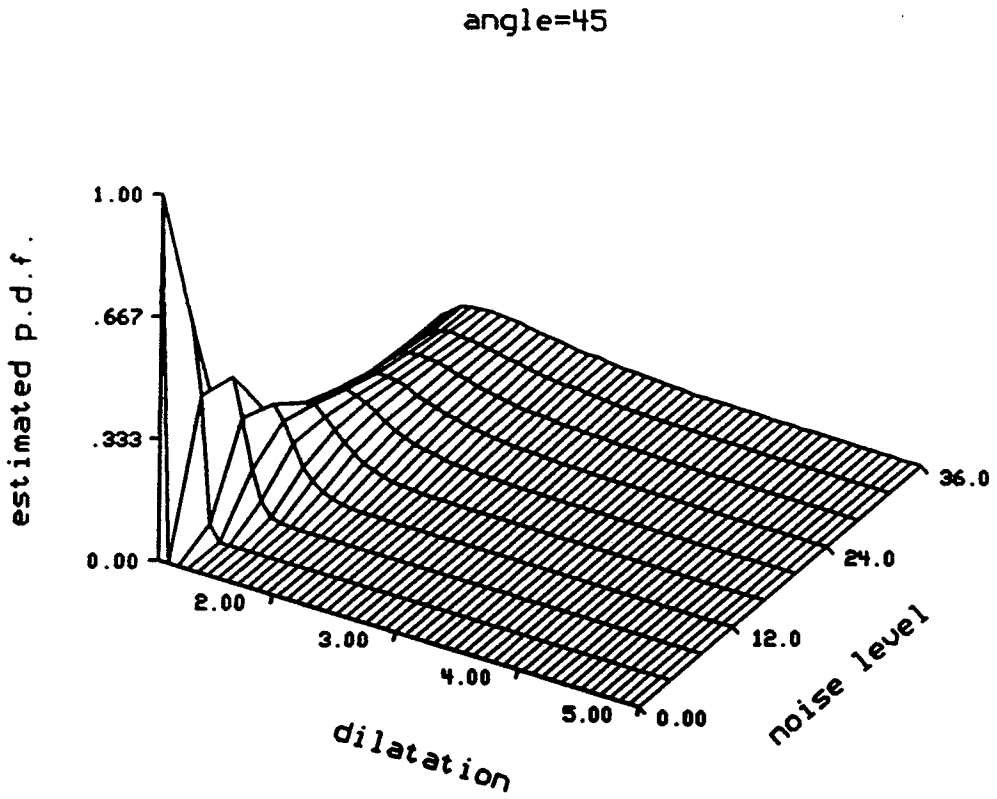


Figure 2.18. Estimated probability density function of the dilatation for angle=45° at various noise levels. Each value of the "noise level" corresponds to a percentage of the smallest perpendicular distance of the triangle. The length corresponding to the percentage value is used as the standard deviation of the zero mean i.i.d. normal random variables used for modeling the distortion in the landmarks.

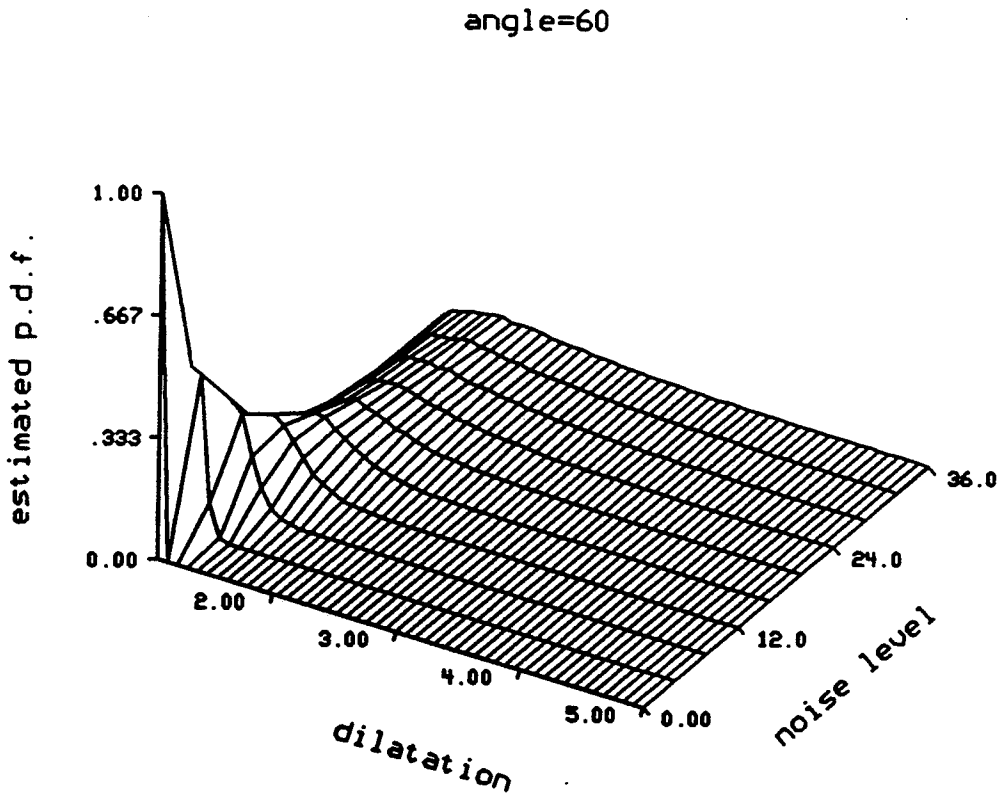


Figure 2.19. Estimated probability density function of the dilatation for angle=60° at various noise levels. Each value of the "noise level" corresponds to a percentage of the smallest perpendicular distance of the triangle. The length corresponding to the percentage value is used as the standard deviation of the zero mean i.i.d. normal random variables used for modeling the distortion in the landmarks.

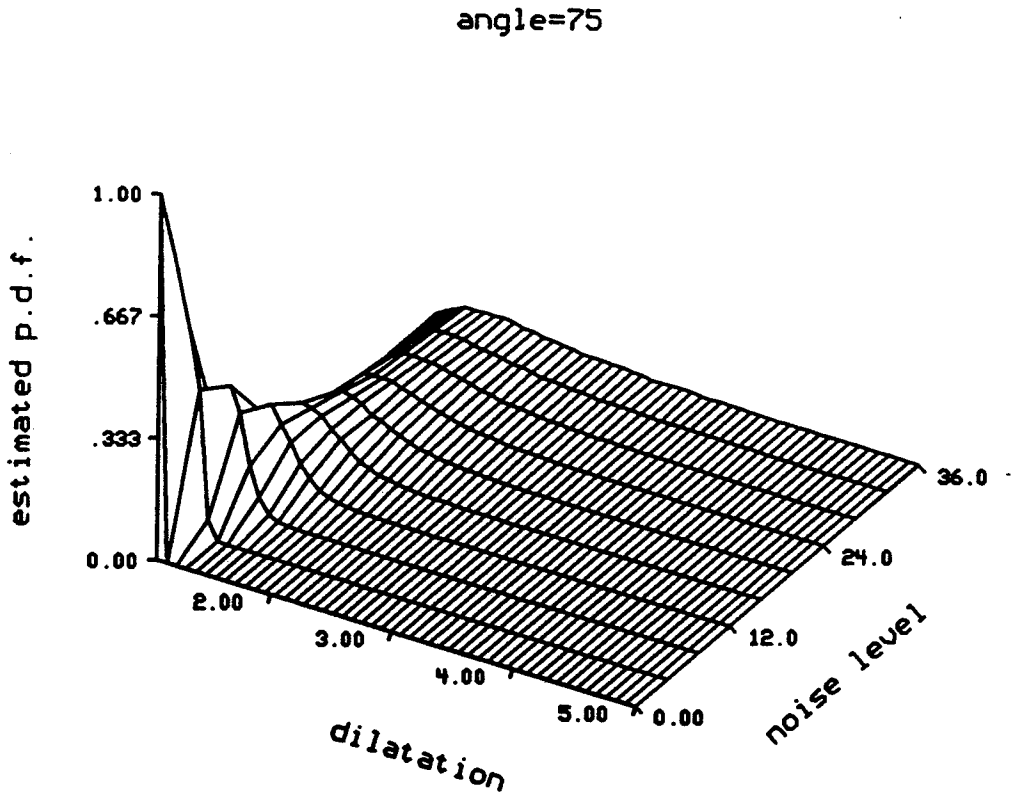


Figure 2.20. Estimated probability density function of the dilatation for angle=75° at various noise levels. Each value of the "noise level" corresponds to a percentage of the smallest perpendicular distance of the triangle. The length corresponding to the percentage value is used as the standard deviation of the zero mean i.i.d. normal random variables used for modeling the distortion in the landmarks.

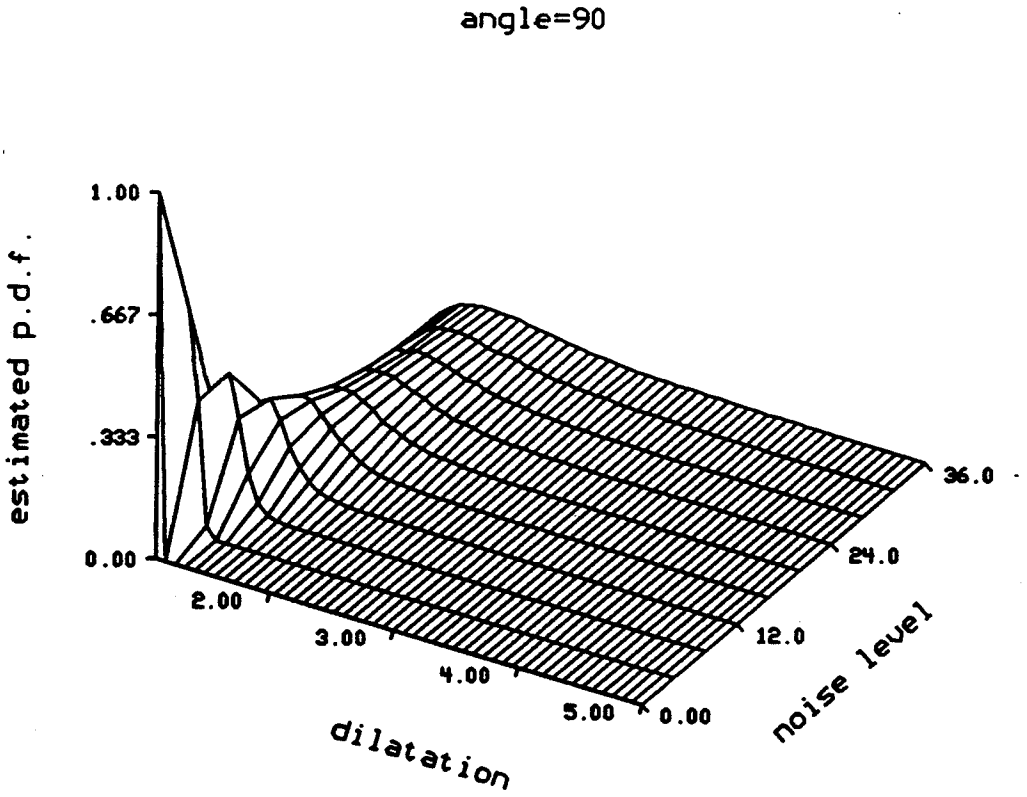


Figure 2.21. Estimated probability density function of the dilatation for angle=75° at various noise levels. Each value of the "noise level" corresponds to a percentage of the smallest perpendicular distance of the triangle. The length corresponding to the percentage value is used as the standard deviation of the zero mean i.i.d. normal random variables used for modeling the distortion in the landmarks.

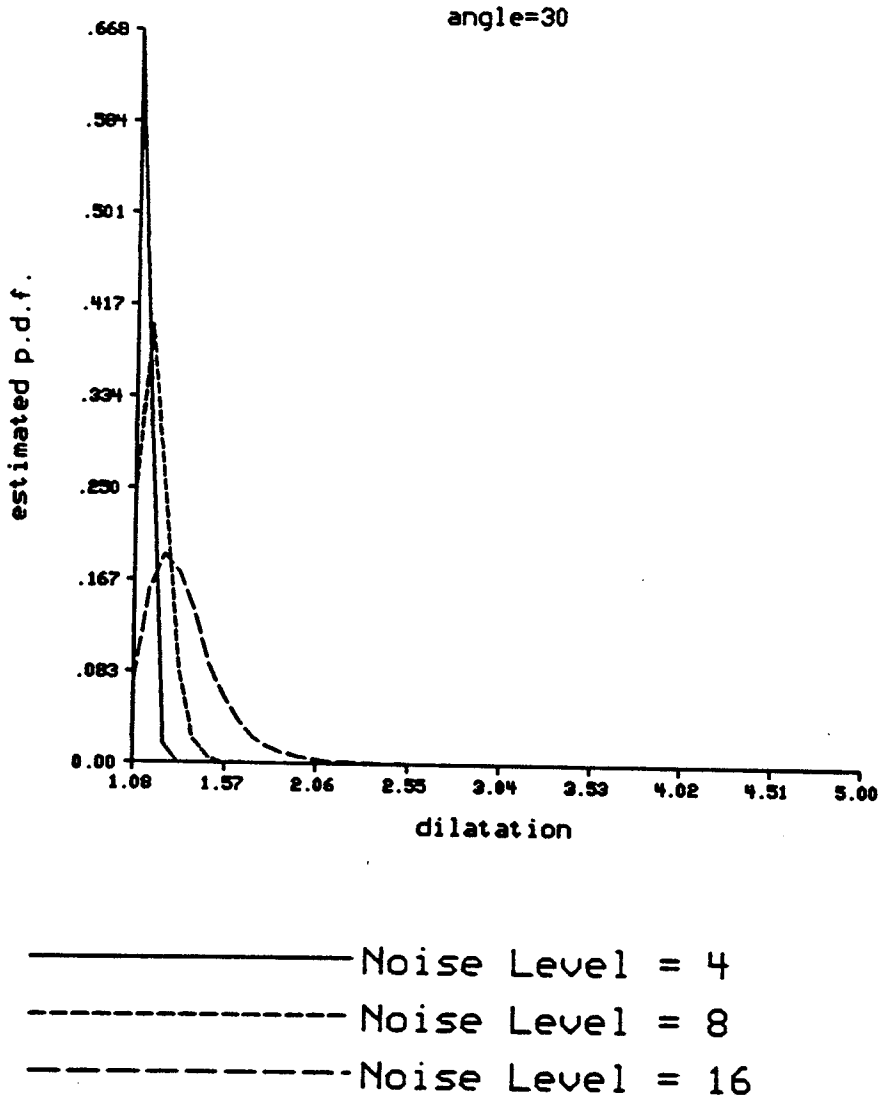


Figure 2.22. Profiles of the plot of Figure 2.17. They correspond to the estimated probability density function of the dilatation for angle=30° at noise levels of 4, 8, and 16.

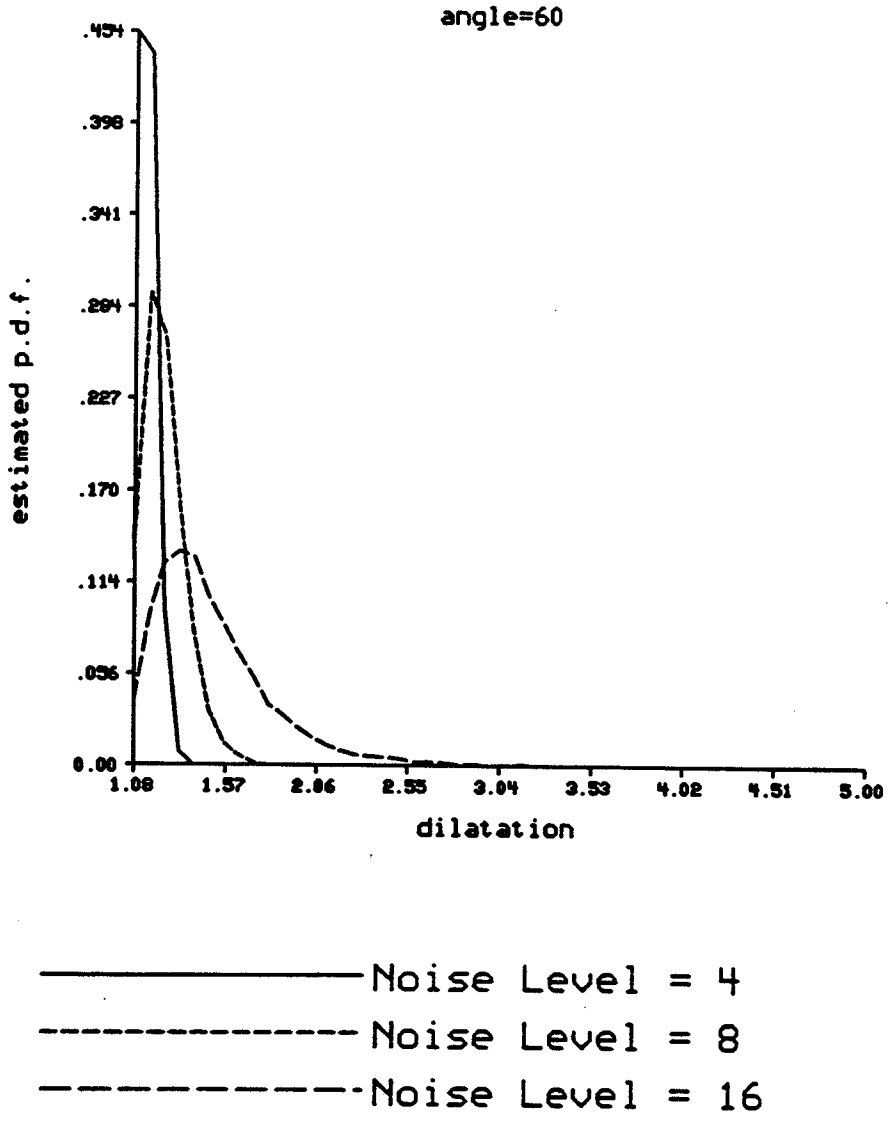


Figure 2.23. Profiles of the plot of Figure 2.19. They correspond to the estimated probability density function of the sphericity for angle=60° at noise levels of 4, 8, and 16.

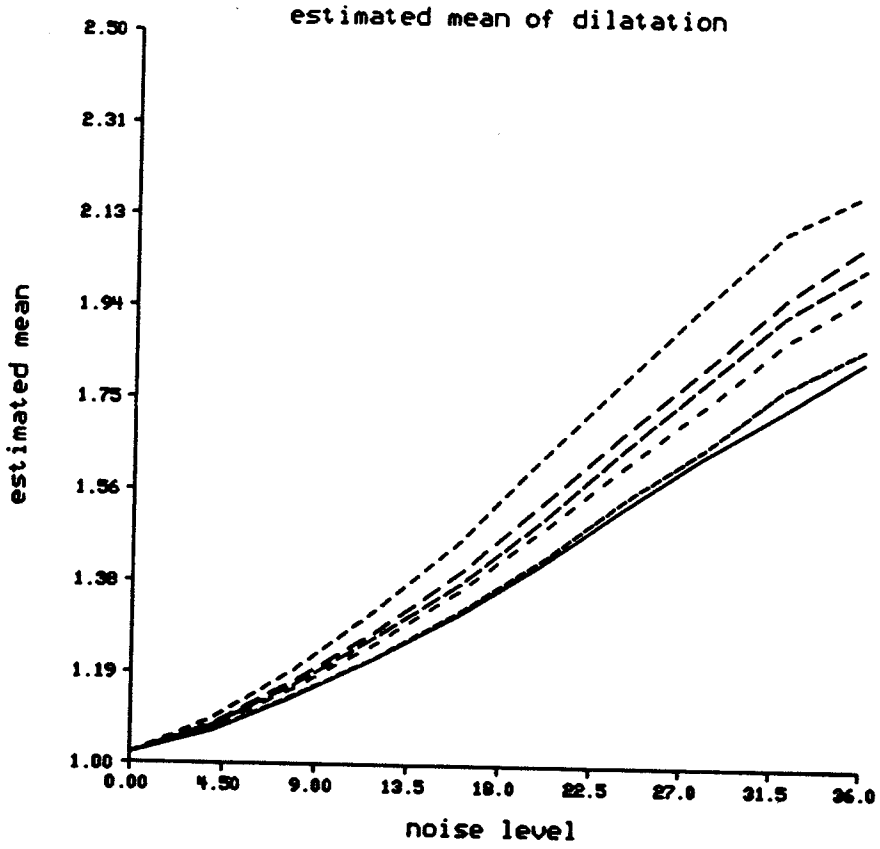


Figure 2.24. Estimated mean of the dilatation corresponding to different types of triangles specified by angles 15°, 30°, 45°, 60°, 75°, and 90°.

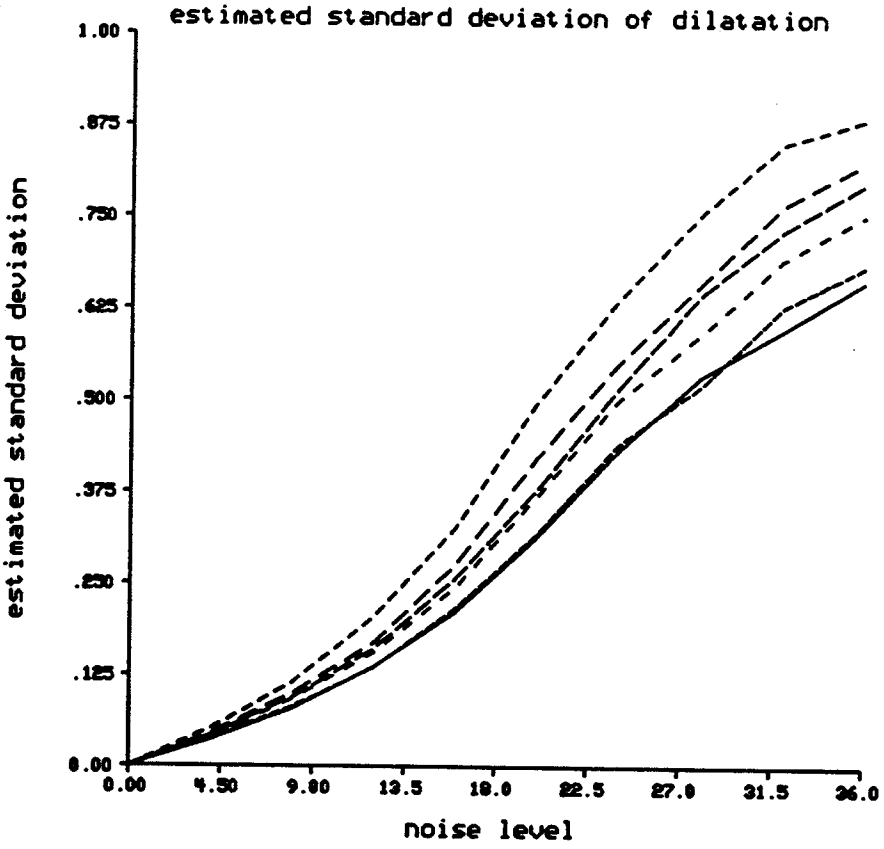


Figure 2.25. Estimated standard deviation of the dilatation corresponding to different types of triangles specified by angles 15°, 30°, 45°, 60°, 75°, and 90°.

CHAPTER 3

LANDMARK EXTRACTION

3.1. Introduction

The shape features of each model in a library as well as those of the objects in a scene have to be extracted before the shape recognition task can be performed. The complexity of the feature extractor depends on the desired shape features, which in turn depend on the nature of the shape recognition algorithm. A shape feature can be classified as either a global or a local shape feature. Examples of global shape features are the silhouette and contour of an object. Examples of shape recognition methods that operate on the entire object silhouette and contour are the moment shape description method [Hu62, Dud77], and the Fourier shape description method [Per77], respectively. Examples of local shape features which represent portions of an object are line segments, edges, and corners of the object. Many partial shape recognition methods [Bha84, Pri84, Aya86, Bha87, Koc87] use line segments as shape features to achieve recognition. This chapter discusses the feature extraction task of the landmark-based shape recognition system shown in Figure 1.4. We refer to this task as landmark extraction.

The landmark-based shape recognition approach uses landmarks, which are local shape features, to recognize objects in a scene. As mentioned earlier, landmarks are points of interest of the object that have important shape attributes. They are usually the extreme points, such as corners, holes, protrusions, and points with high curvature. They can also be problem specific based on *a priori* knowledge. It is important to note that the entire object contour of an object is *not* needed for this approach to achieve recognition; the approach only requires knowledge of the positions of the landmarks of the object in the image. It is necessary to order the landmarks as consecutive points along the object boundary. However, if the interior points of the object are used as landmarks, it is necessary to arrange them in a pre-defined order reflecting the shape and geometry of the object.

Among the extreme points, points with high curvature along the object contour are features that are most attractive. An object contour is the boundary of the object. This 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. In this chapter, we shall consider landmarks as the points of high curvature on an object contour. Other problem specific types of landmarks will not be considered. Note that erroneous landmarks of objects in a scene may occur due to object occlusion or noise in the scene. The effectiveness of the landmark matching task with respect to erroneous scene landmarks will be discussed in the next two chapters. For illustrative purposes, we assume that images are obtained by orthographic projection, and the silhouette of an object region is either given or can be easily acquired from an back-lit image by a simple thresholding operation. Examples of back-lit images of objects together with their corresponding boundaries are shown in Figures 3.1-3.5. The boundary of each object region is extracted by means of a chain code [Fre74].

Many partial shape recognition methods use line segments which are derived from polygonal approximation of an object contour as shape features. The vertices of the approximated polygon are usually points with high curvature along the object contour. The vertices are also known as the break points. We shall begin by discussing two commonly used polygonal approximation algorithms [Ram72, Pav74] in Section 3.2. We shall then present two methods of detecting landmarks from contours, the curvature guided polygonal approximation method and the cardinal curvature points method, in Section 3.3 and Section 3.4, respectively. Finally, a summary of this chapter will be given in Section 3.5.

3.2. Polygonal Approximation

Polygonal approximation is the representation of an object boundary by a polygon. It has been used to extract line segments used as shape features for many partial shape recognition algorithms [Bha84, Pri84, Aya86, Bha87, Koc87]. Two commonly used polygonal approximation approaches will be discussed. The first approach is Ramer's algorithm [Ram72]. The second approach is a split-and-merge algorithm similar to the one discussed in [Pav74]. A common function shared by most polygonal approximation algorithms is the collinearity test that checks if points along a boundary portion are collinear with respect to a straight line. Collinearity is usually determined by the maximum perpendicular distance from a point of the boundary portion to the straight

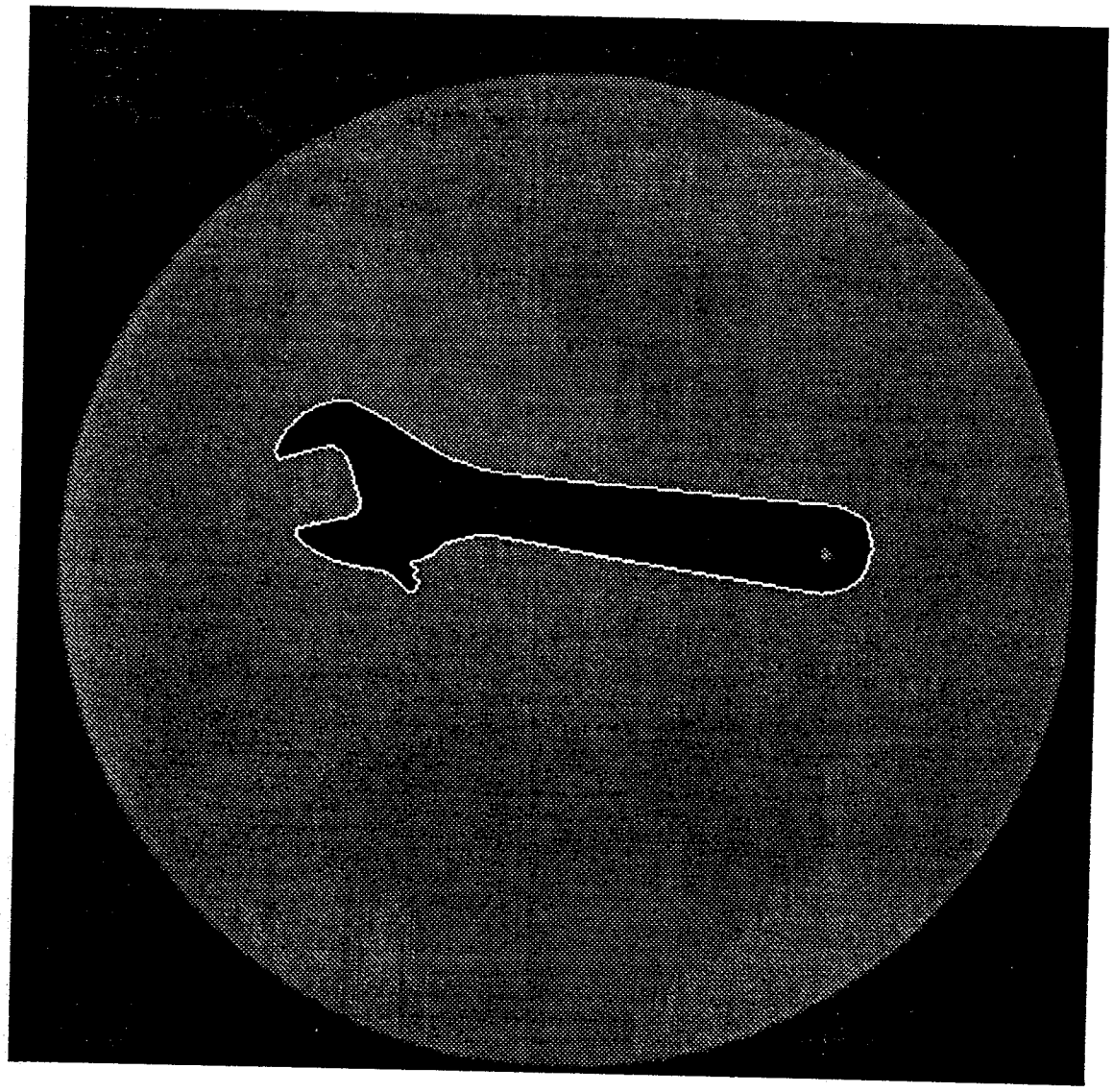


Figure 3.1. A 512×512 back-lit image of a wrench together with its boundary.

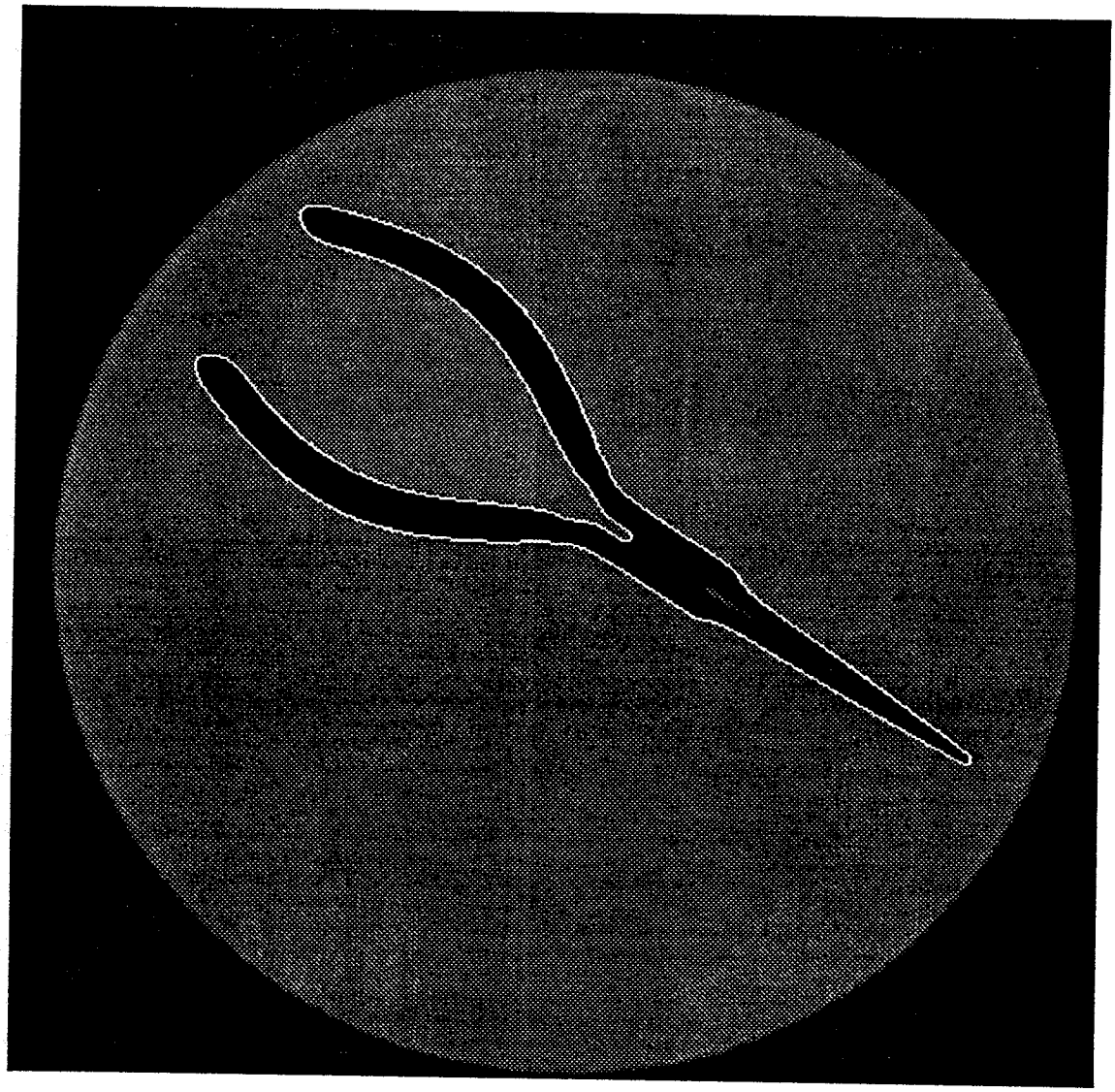


Figure 3.2. A 512×512 back-lit image of a needle-nose plier together with its boundary.

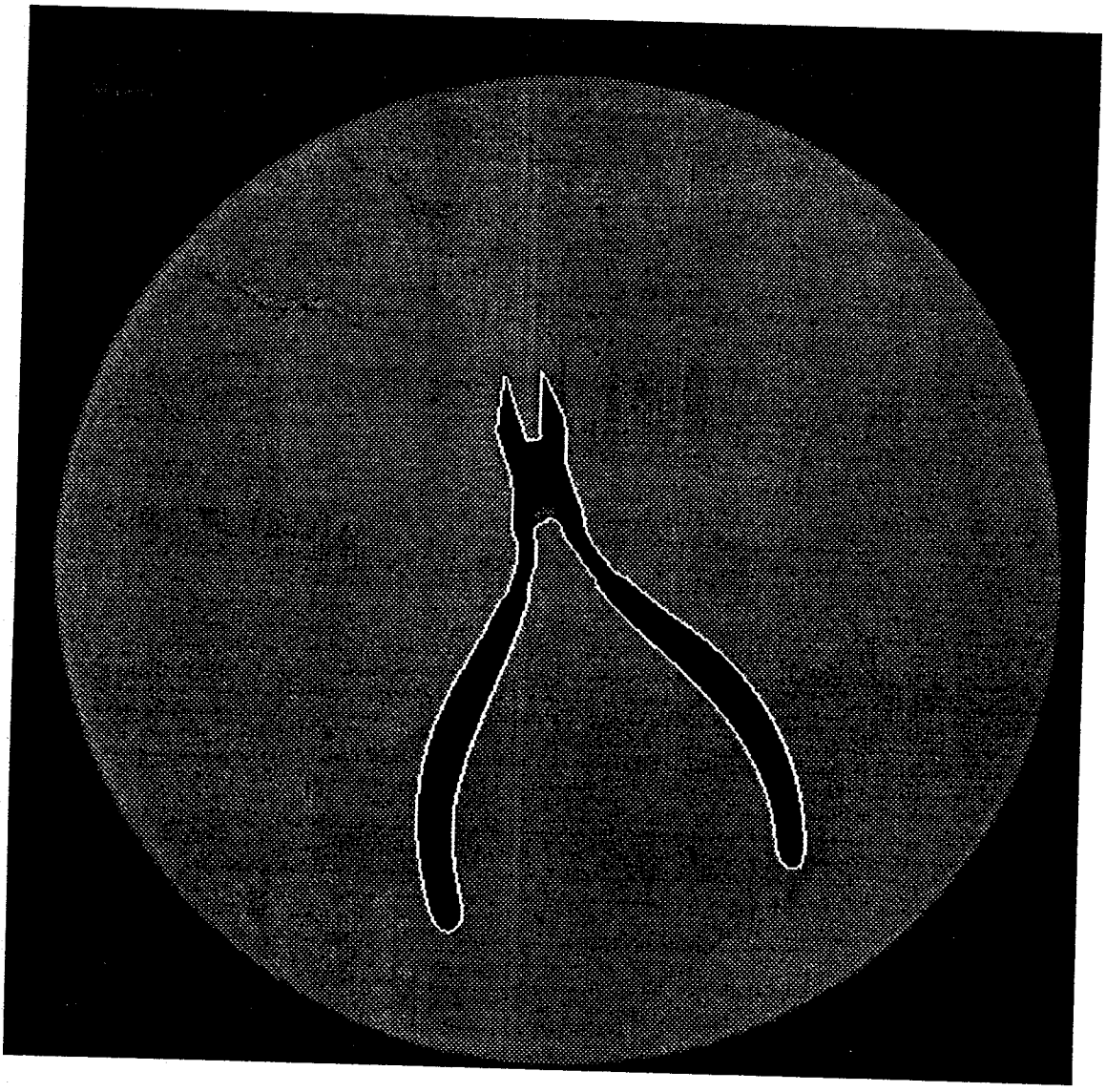


Figure 3.3. A 512×512 back-lit image of a wire cutter together with its boundary.

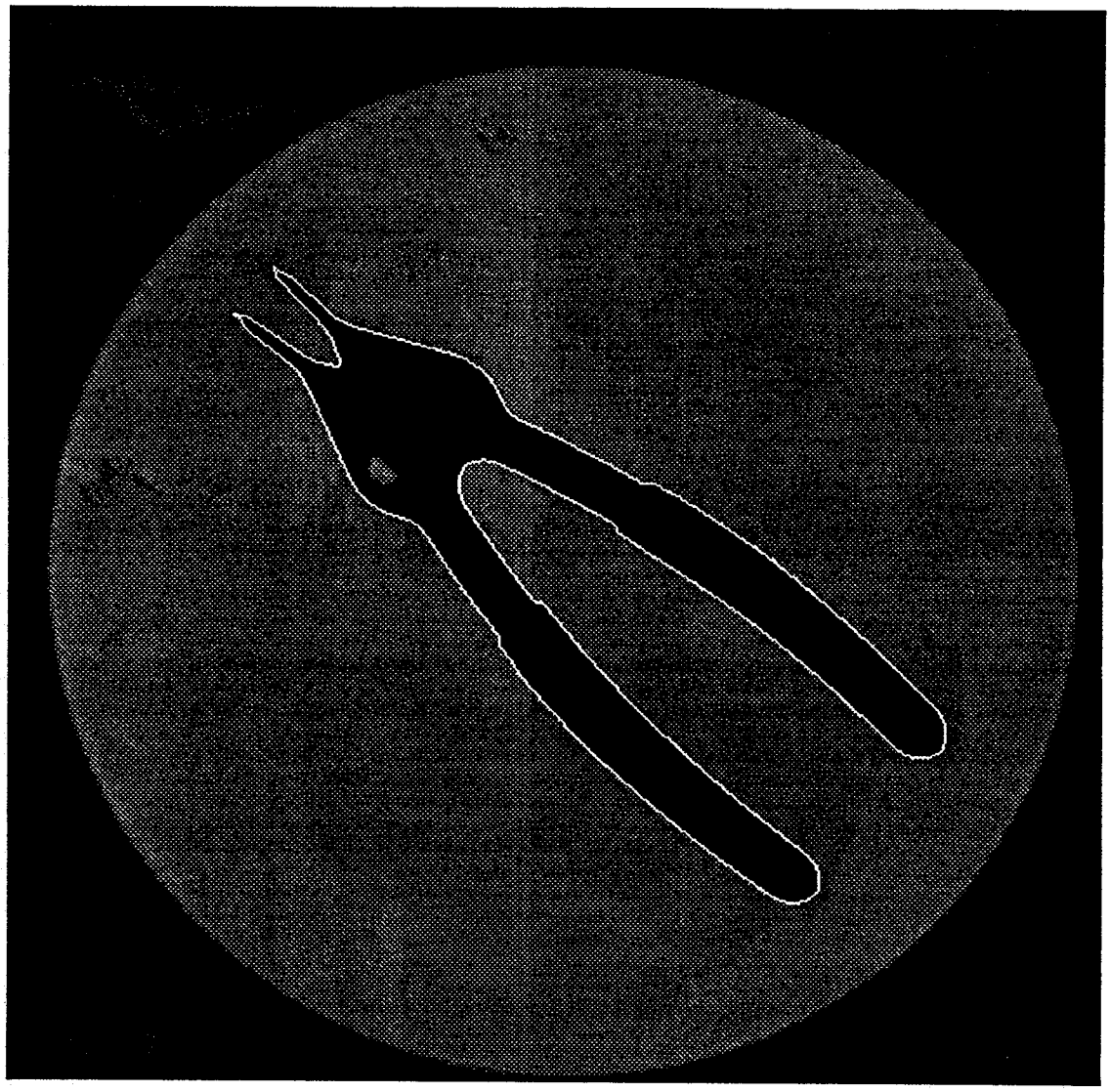


Figure 3.4. A 512×512 back-lit image of a specialty plier together with its boundary.

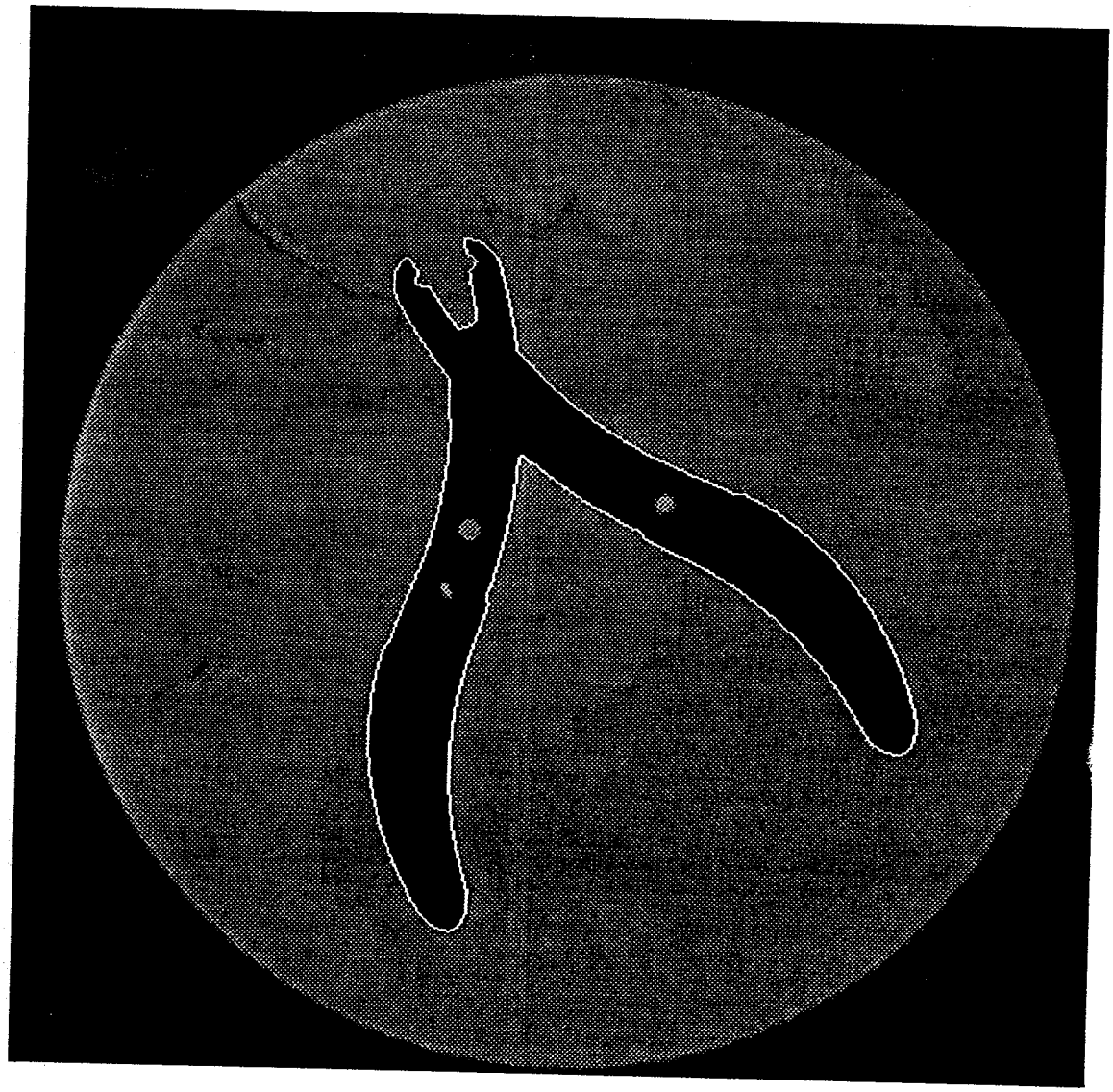
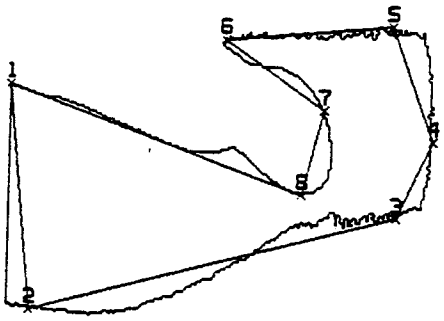


Figure 3.5. A 512×512 back-lit image of a wire stripper together with its boundary.

line. Consider a boundary portion between two points, A and B, we compute the maximum perpendicular distance from the boundary portion to the straight line AB. If the distance is within tolerance, that boundary portion is approximated by the straight line segment AB. Otherwise, the point along the boundary portion that yields the maximum distance becomes a new break point, say C, and the boundary portion is approximated by two line segments AC and CB.

Ramer's algorithm finds a set of break points along a given closed or open boundary using the above collinearity criterion repeatedly. The boundary is then approximated with a polygon by joining the break points with straight lines. Consider a closed boundary, Ramer suggests that the top left most point and the bottom right most point of the boundary be used as the two initial break points. Based on these initial break points, new break points along the boundary are iteratively determined.

We shall illustrate the algorithm by means of the following example:



The two initial break points are point 1 which is the top left most point, and point 4 which is the bottom right most point of the contour. Along the upper right portion of the boundary between the two initial break points, point 5 has the maximum perpendicular distance from the boundary portion to the straight line joined by points 1 and 4. It is greater than a given tolerance, and hence becomes a new break point. Similarly, point 2 along the lower left portion of the boundary between the two initial break points becomes a new break point. The maximum perpendicular distance from the boundary portion between points 4 and 5 is within tolerance, and hence no new break point is found along this boundary portion. Along the boundary portion between points 1 and 5, point 8 which yields the maximum perpendicular distance to the straight line between points 1 and 5 that is greater than the tolerance becomes

a new break point. Continuing in this fashion for the remaining portions of the boundary that have not passed the collinearity test, new break points along the boundary are found as shown. The algorithm stops when no more break point can be found in which case all break points have passed the collinearity test.

Figure 3.6a shows a closed boundary extracted from a 512×512 image. The boundary at a different orientation and a different scale are shown in Figures 3.7a and 3.8a, respectively. Applying Ramer's algorithm to these boundaries results in the approximated polygons shown in Figures 3.6b, 3.7b, and 3.8b, respectively. A tolerance of 20 pixels is used for the algorithm. These examples show that the number of break points and their locations along the boundary obtained by Ramer's algorithm are not the same if the same boundary is at a different orientation or a different scale. That is, break points along the boundary obtained by this algorithm are sensitive to the orientation and the scale of the boundary.

The split-and-merge algorithm requires a slightly more complex procedure than Ramer's algorithm. There are several versions of the split-and-merge algorithm. The one similar to [Pav74] will be discussed. Pavlidis and Horowitz [Pav74] approximate boundary points by interpolating straight line segments. Consequently, the break points of the approximated polygon do not usually lie on the original boundary, and in fact, can be far from it. In contrast, the following split-and-merge algorithm will find break points along a given boundary. It is briefly outlined below:

(1) Assign an arbitrary number of points along the boundary as the initial set of break points. The initial approximated polygon is formed by joining the sequence of break points along the original boundary with straight lines.

(2) For each pair of adjacent break points, determine the point along the boundary portion that yields the maximum perpendicular distance to the straight line segment joined by the two break points. If the maximum perpendicular distance is greater than a given tolerance, that point becomes a new break point; i.e., the line segment is replaced by two line segments. This is the "splitting" part of the algorithm.

(3) For each pair of adjacent line segments comprising of three consecutive break points, say A, B, and C, compute the maximum perpendicular distance from the boundary portion between A and C to line AC. If the distance is within tolerance, break point B is removed. That is, line segments AB and BC are replaced by line segment AC. Note that each

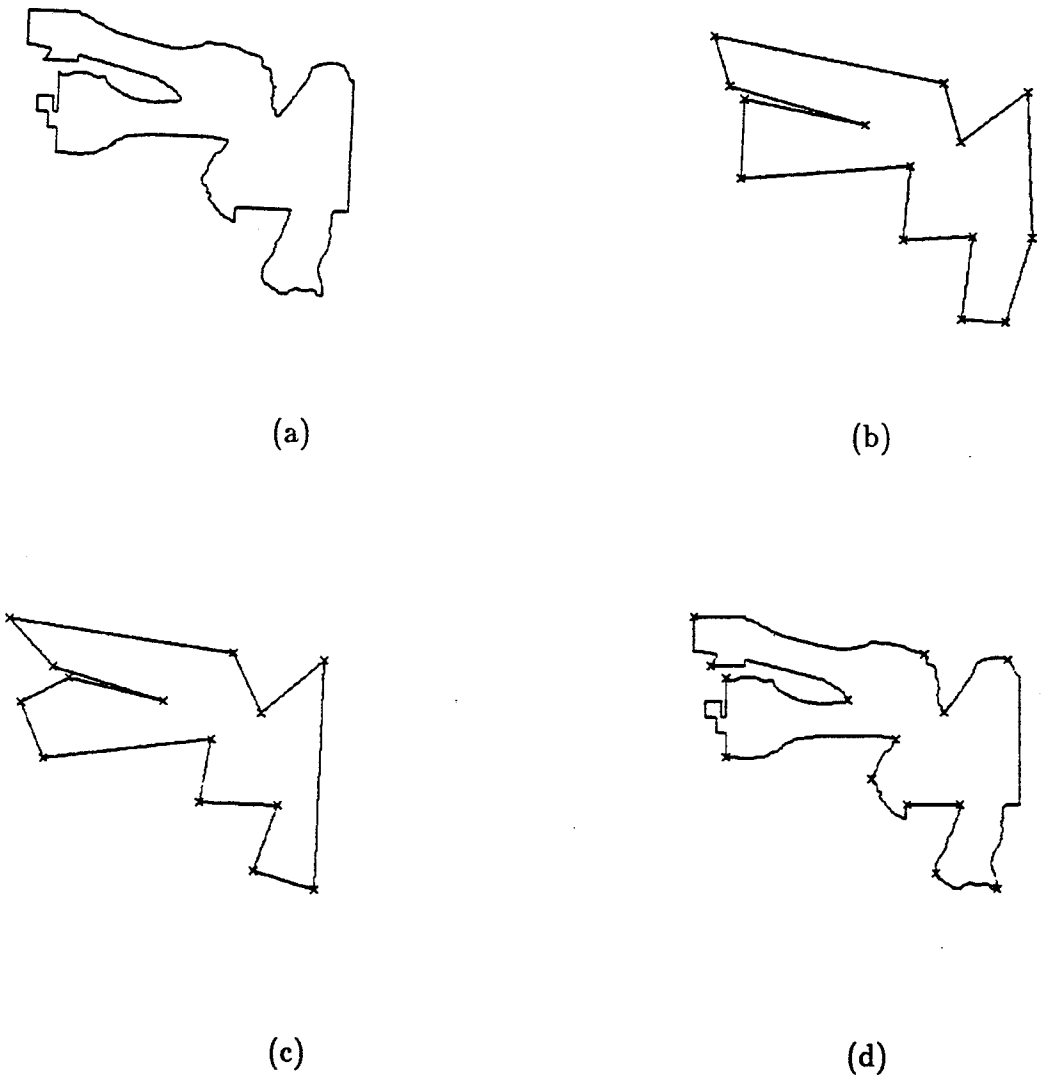


Figure 3.6. Results of polygonal approximation of a contour using different methods. (a) A contour extracted from a 512×512 image. (b) The approximated polygon of the contour obtained by Ramer's algorithm. (c) The approximated polygon of the contour obtained by the split-and-merge algorithm. (d) The landmarks of the contour obtained by curvature guided polygonal approximation. Each landmark is indicated by an "×."

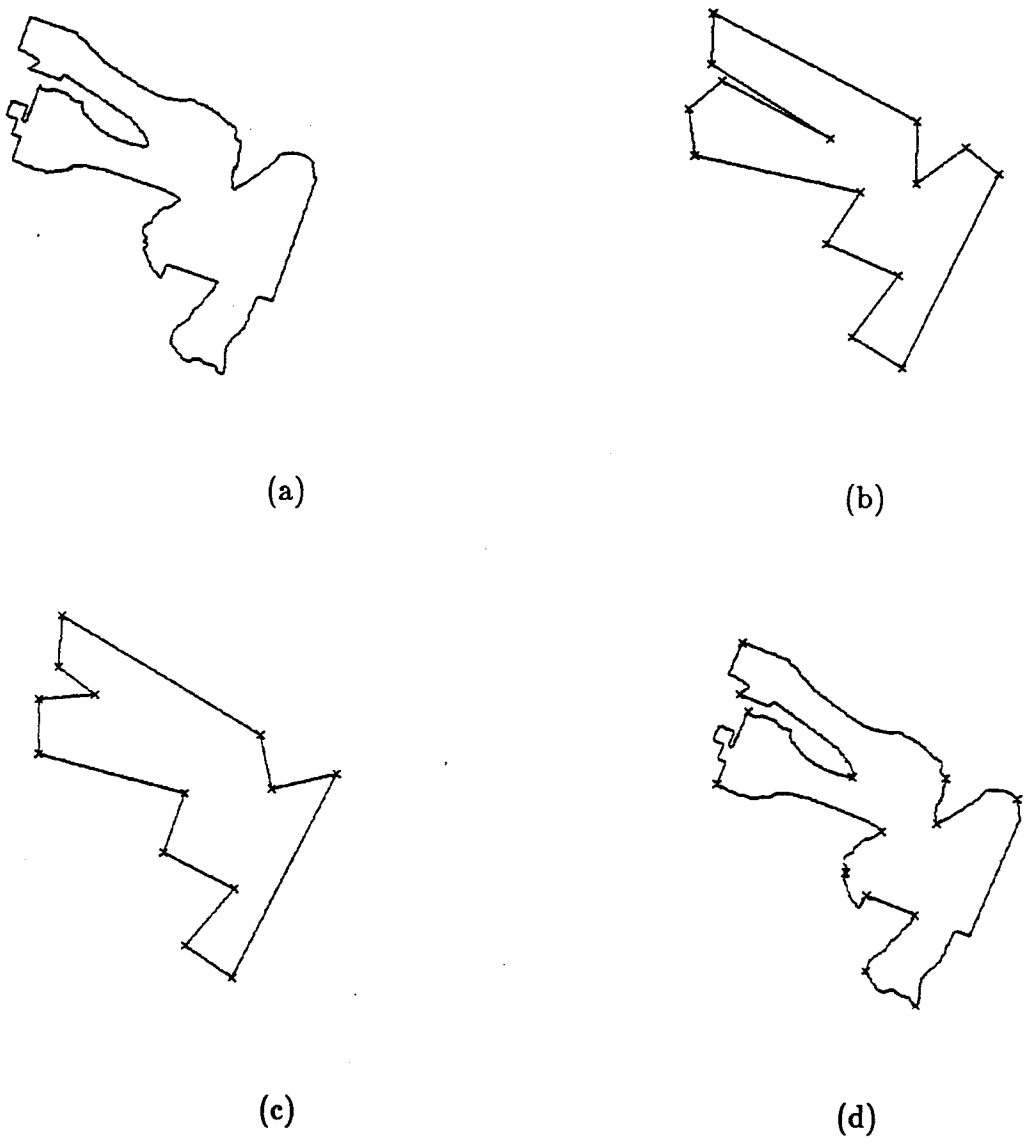


Figure 3.7. Results of polygonal approximation of the rotated contour using different methods. (a) The rotated contour of Figure 3.6a. (b) The approximated polygon of the rotated contour obtained by Ramer's algorithm. (c) The approximated polygon of the rotated contour obtained by the split-and-merge algorithm. (d) The landmarks of the rotated contour obtained by curvature guided polygonal approximation. Each landmark is indicated by an "×."

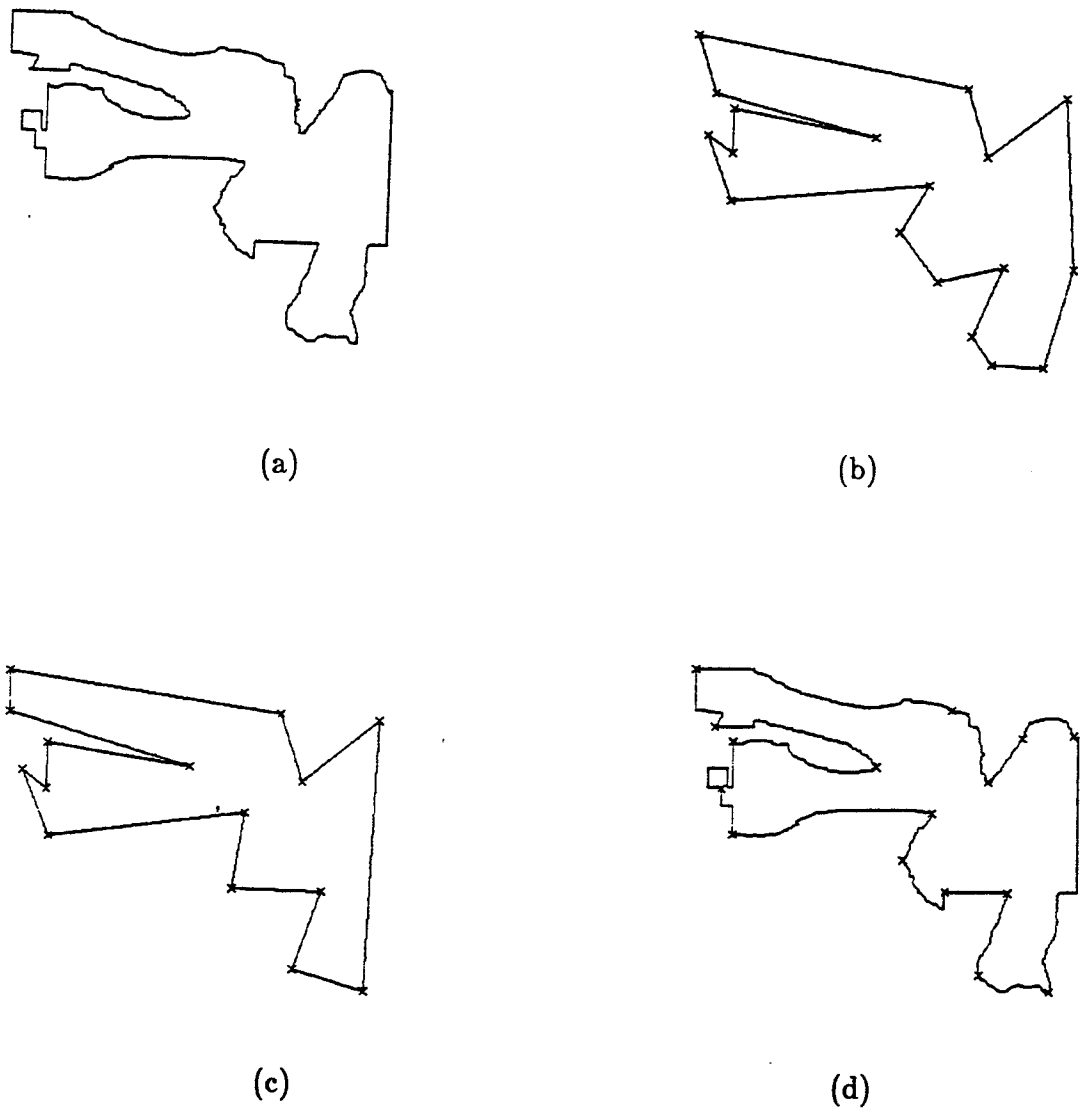


Figure 3.8. Results of polygonal approximation of the enlarged contour using different methods. (a) The enlarged contour of Figure 3.6a. (b) The approximated polygon of the enlarged contour obtained by Ramer's algorithm. (c) The approximated polygon of the enlarged contour obtained by the split-and-merge algorithm. (d) The landmarks of the enlarged contour obtained by curvature guided polygonal approximation. Each landmark is indicated by an "×."

replacement is immediately tested for merging with the next line segment. This is the "merging" part of the algorithm.

(4) Repeat Steps (2) and (3) until an equilibrium is reached; i.e., no more splitting and merging is necessary.

Applying the split-and-merge algorithm to the same set of boundaries that are used for Ramer's algorithm results in the approximated polygons shown in Figures 3.6c, 3.7c, and 3.8c. Ten equally spaced points along the boundary are assigned as the initial set of break points. As in Ramer's algorithm, a tolerance of 20 pixels is again used. Like Ramer's algorithm, the number of break points and their locations along the boundary obtained by the split-and-merge algorithm are not the same if the boundary is at a different orientation or a different scale.

Both algorithms are very sensitive to the tolerance. The original starting set of break points will also affect the final result. Regardless of the scale and the orientation of an object, the number of landmarks of the object should not vary, and their locations relative to the object should not deviate. Break points obtained by the above polygonal approximation algorithms that vary with the orientation and the scale of the boundary are thus not desirable landmarks. Two approaches that obtain more stable break points are discussed in the following sections.

3.3. Curvature Guided Polygonal Approximation

As mentioned above, different starting sets of break points used in a polygonal approximation algorithm will result in different approximated polygons. A good starting set of break points is thus important to the polygonal approximation algorithm. Points with extreme curvature are potential break points for approximating a boundary with a polygon. Since these extreme curvature points are likely to be break points, we propose to use these points as a starting set of break points for the polygonal approximation algorithm. The split-ang-merge algorithm will be used as the polygonal approximation algorithm to correct and modify the original starting set of break points. We shall call such an approach *curvature guided polygonal approximation.*

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 these false concavities and convexities. It has been shown that a Gaussian filter is an ideal smoothing filter for numerical differentiation [Tor86]. We use the approach of smoothing a planar curve with

a Gaussian filter to find a starting set of break points for the polygonal approximation. A planar curve can be represented by a set of points in parametric form,

$$(x(t),\, y(t)) \in \mathbb{R}^2 \quad ,$$

where t is the path length along the curve. Smoothing the curve with a Gaussian filter is the same as convolving $x(t)$ and $y(t)$, respectively, with a one-dimensional Gaussian filter,

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

where ω is the width (spatial support) of the filter. Denote the Gaussian smoothed curve by the set of points $(X(t,\, \omega),\, Y(t,\, \omega))$. That is,

$$X(t,\, \omega) = x(t) \,*\, \eta(t,\, \omega) \quad , \tag{3.1}$$

$$Y(t,\, \omega) = y(t) \,*\, \eta(t,\, \omega) \quad , \tag{3.2}$$

where $*$ indicates convolution. It can be shown using elementary calculus [Tho72] (as was shown in [Mok86]) that the curvature of the smoothed curve is:

$$\kappa(t,\, \omega) = \frac{\dot{X}\ddot{Y} - \dot{Y}\ddot{X}}{(\dot{X}^2 + \dot{Y}^2)^{3/2}} \quad , \tag{3.3}$$

where t is the path length along the curve,
ω is the width of the Gaussian filter,
κ is the curvature of the curve at t,

$$\dot{X} = \frac{dX}{dt} \quad , \quad \ddot{X} = \frac{d^2X}{dt^2} \quad ,$$

$$\dot{Y} = \frac{dY}{dt} \quad , \quad \ddot{Y} = \frac{d^2Y}{dt^2} \quad .$$

Note that the arguments of $X(t,\, \omega)$ and $Y(t,\, \omega)$ have been dropped. As we "traverse" along t in increasing values of t, a positive curvature corresponds to a concavity on our left, and a negative curvature corresponds to a concavity on our right. We therefore propose to select points along a curve that corresponds to the positive maximum and the negative minimum curvature points of the Gaussian smoothed curve as the starting set of break points for polygonal approximation. From now on, we refer to an extreme curvature point as either a positive maximum or a negative minimum curvature point.

The curvature guided polygonal approximation algorithm as applied to a closed boundary can be summarized by the following procedure:

(1) Remove all one-pixel wide protrusions. Figure 3.9 shows an example of a one-pixel wide protrusion which may result due to the discrete boundary representation and quantization error.

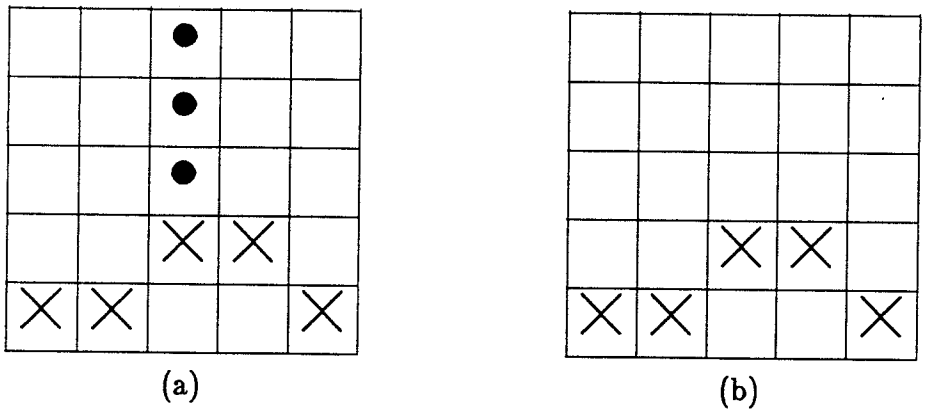


Figure 3.9. An example of a one-pixel wide protrusion. (a) One-pixel wide protrusion. (b) After removing the protrusion. × indicates a boundary pixel. • indicates a boundary protrusion pixel.

(2) Smooth the boundary with the above Gaussian filter.

(3) Find the set of positive maximum and negative minimum curvature points along the Gaussian smoothed boundary.

(4) The points along the original boundary (with one-pixel wide protrusions removed) that correspond to the set of points found in Step (3) are used as the starting set of break points for polygonal approximation of the original boundary.

(5) Employ the split-and-merge polygonal approximation algorithm mentioned in Section 3.2.

(6) The resulting break points are the landmarks of the boundary.

Two parameters, ω of the Gaussian filter and the tolerance for collinearity, must be set in using the algorithm. There is a trade-off when choosing the value of ω. A large value will remove small details of the boundary curvature, while a small value will permit false concavities and convexities. Since we only want to estimate a starting set of break points for the split-and-merge polygonal approximation algorithm, a flexible range of

values for ω is feasible. The tolerance is scale dependent; i.e., using the same tolerance for a boundary at a different scale may yield a different result. It is usually chosen by trial and error, or based on *a priori* knowledge about the scale of the boundary. The curvature guided polygonal approximation algorithm is thus sensitive to scaling. It does, however, eradicate two undesirable characteristics associated with most polygonal approximation algorithms; it provides a good starting set of break points, and it is less sensitive to orientation.

The results of applying curvature guided polygonal approximation algorithm to the same set of boundaries that are used for Ramer's algorithm are shown in Figures 3.6d, 3.7d, and 3.8d, respectively. A width of 5 pixels for the Gaussian filter and a tolerance of 20 pixels for the collinearity test are used. The examples indicate that break points obtained by this algorithm are less sensitive to orientation, but remains sensitive to scaling.

A step by step pictorial depiction of extracting the landmarks of a "wrench" image (Figure 3.1) using curvature guided polygonal approximation algorithm is described below. Figure 3.10 shows the Gaussian smoothed contour of Figure 3.1 with ω=10. The corresponding curvature function is shown in Figure 3.11, where a "*" symbol indicates a break point which is either a local positive maximum or a local negative minimum curvature point. Only points on straight line segments have curvature value of zero. Since extreme curvature points with curvature values close to zero are likely lying on curve segments that are almost straight, we consider only those positive maxima which lie above a specified threshold. Likewise we consider only those negative minima which lie below another specified threshold. From empirical results, we have found that using 0.0035 and -0.0035 as the respective positive and negative threshold provides reasonable results. The initial break points along the original boundary are shown in Figure 3.12. After the split-and-merge polygonal approximation using a tolerance of 15 pixels, the final set of landmarks along the original boundary are shown in Figure 3.13. Using the same parameters as above, the extracted landmarks of the needle-nose plier (Figure 3.2), the wire cutter (Figure 3.3), the specialty plier (Figure 3.4), and the wire stripper (Figure 3.5) are shown in Figures 3.14-3.17, respectively.

Figure 3.10. The Gaussian smoothed boundary of a wrench (Figure 3.1) using ω=10. Each "×" indicates an extreme curvature point.

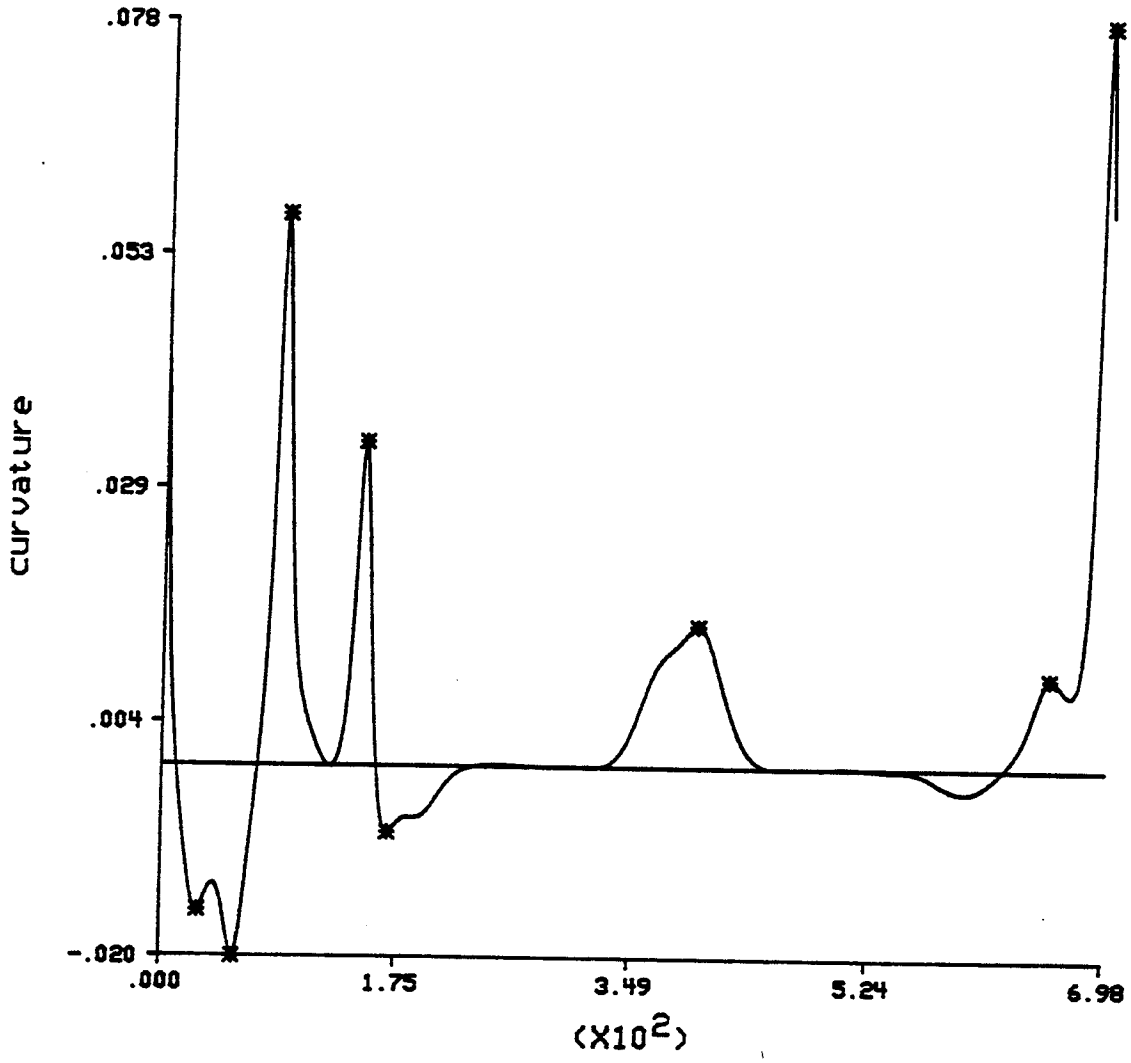


Figure 3.11. The curvature function of the Gaussian smoothed boundary of the wrench using ω=10. Each "*" indicates an extreme curvature point.

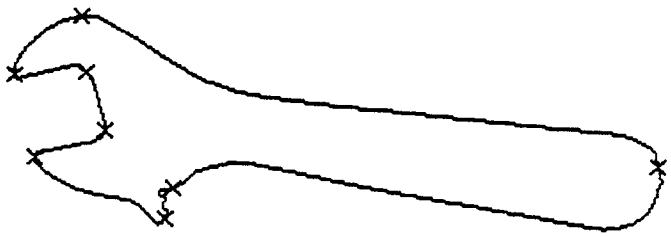


Figure 3.12. The initial set of break points, each indicated by an "×," used for a subsequent polygonal approximation.

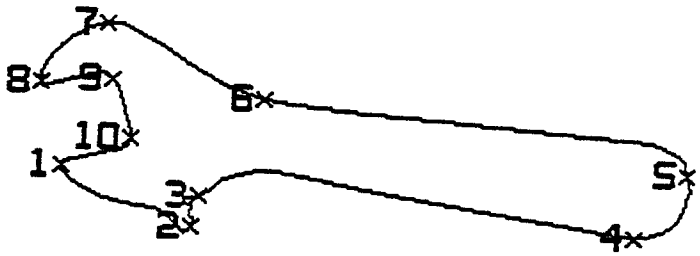


Figure 3.13. The landmarks of the wrench obtained by the curvature guided polygonal approximation. Each landmark is numerically labeled, and is indicated by an "×."

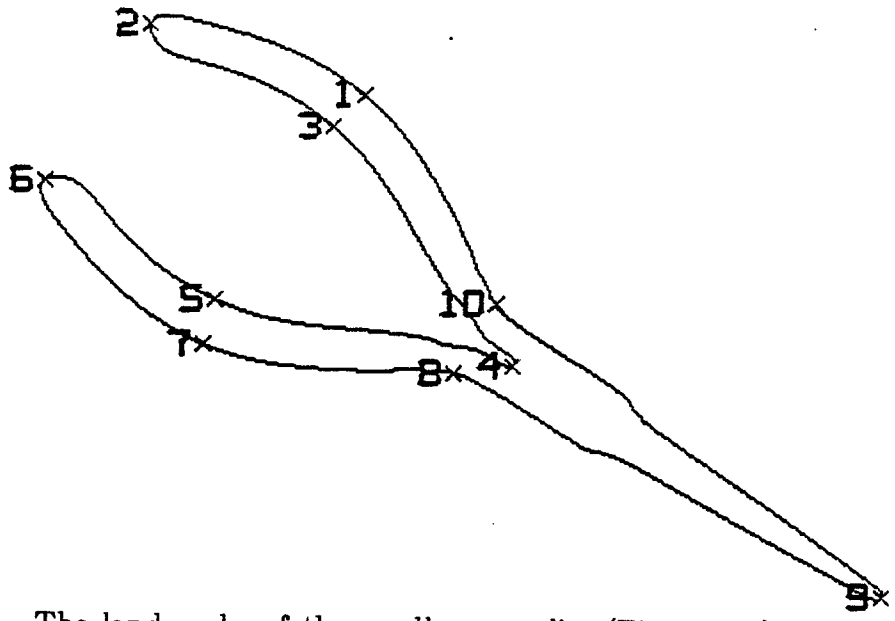


Figure 3.14. The landmarks of the needle-nose plier (Figure 3.2) obtained by the curvature guided polygonal approximation. Each landmark is numerically labeled, and is indicated by an "×."

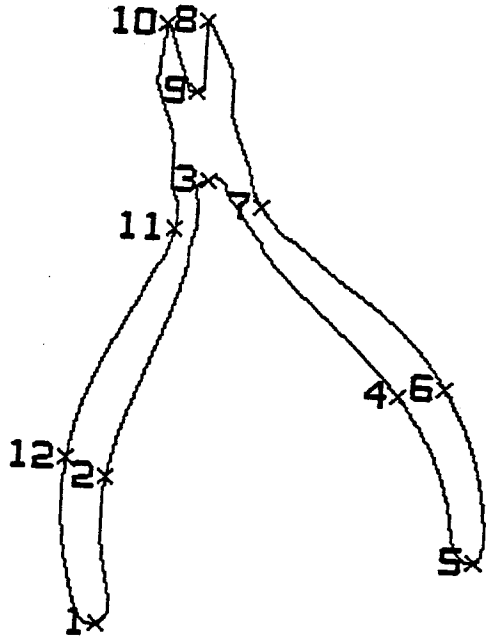


Figure 3.15. The landmarks of the wire cutter (Figure 3.3) obtained by the curvature guided polygonal approximation. Each landmark is numerically labeled, and is indicated by an "×."

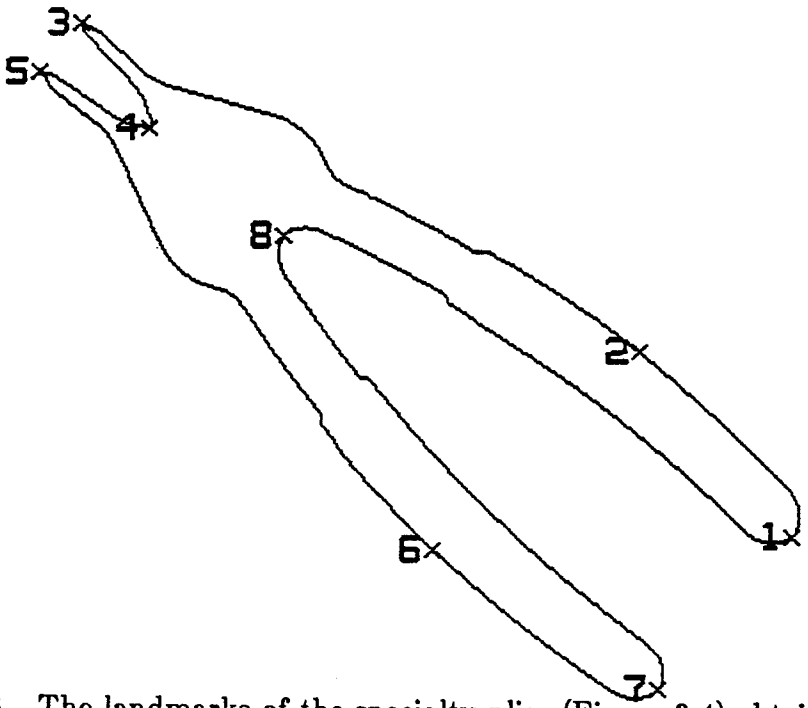


Figure 3.16. The landmarks of the specialty plier (Figure 3.4) obtained by the curvature guided polygonal approximation. Each landmark is numerically labeled, and is indicated by an "×."

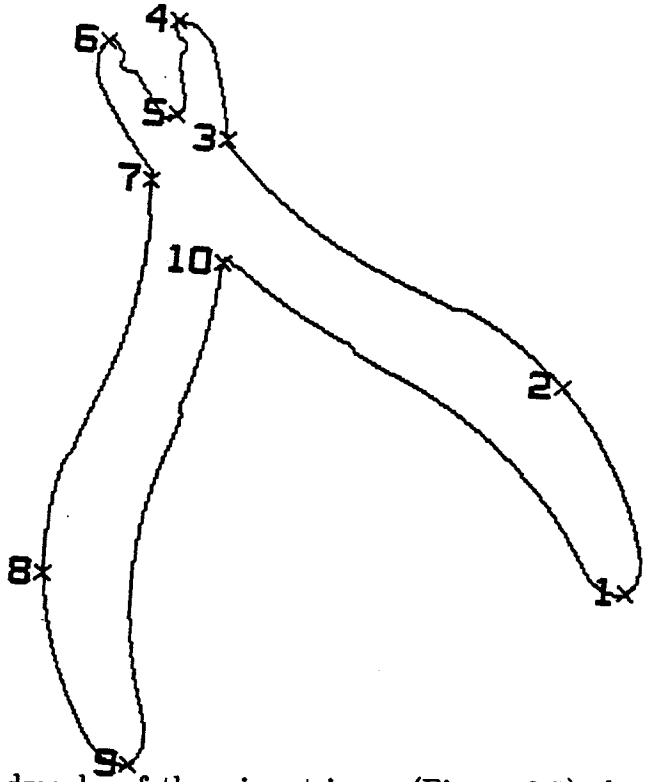


Figure 3.17. The landmarks of the wire stripper (Figure 3.5) obtained by the curvature guided polygonal approximation. Each landmark is numerically labeled, and is indicated by an "×."

3.4 Cardinal Curvature Points

In the curvature guided polygonal approximation algorithm, if the boundary is smoothed by a Gaussian filter with a large ω, false local concavities and convexities are unlikely because of the smoothness of the boundary. The extreme (positive maximum and negative minimum) curvature points of such a smoothed boundary are stable with respect to orientation and scaling; i.e., their locations along the unsmoothed boundary remains relatively unchanged when the boundary is rotated, or scaled within a reasonable range. We shall refer to these stable local extreme curvature points as the *cardinal curvature points.*

The cardinal curvature points along the boundary of an object are suitable landmarks of the object. They are extreme curvature points of the boundary that are stable with respect to Gaussian smoothing for a reasonable range of values of ω and, at the same time, possess the shape attributes of the boundary. Since ω determines the degree of detail (smoothing) of the smoothed boundary, stability of the cardinal curvature points for a reasonable range of values of ω implies that the cardinal curvature points are stable with respect to a reasonable degree of scaling. Since cardinal curvature points are obtained by Gaussian smoothing with a larger ω than the ω used for the curvature guided polygonal approximation, the number of landmarks obtained from the cardinal curvature points is usually less than that by the curvature guided polygonal approximation.

Given a library of model objects, the cardinal curvature points of each object boundary are obtained by successively smoothing the boundary with a Gaussian filter with various widths until the extreme curvature points do not change (their number remains the same, and their locations deviate only a small amount) for a reasonable range of ω. Figures 3.18-3.33 depict the extreme curvature points of the Gaussian smoothed boundaries of a wire cutter (Figure 3.3) and the corresponding curvature functions at various degrees of smoothing. The extreme curvature points of the wire cutter shown in Figure 3.23 are the cardinal curvature points. They are stable for ω ranging from 16 to 28.5. The locations of the cardinal curvature points along the original boundary are the locations of the landmarks of the wire cutter, as shown in Figure 3.34. The landmarks obtained for other objects by this approach are shown in Figures 3.35-3.38, and their corresponding range of stability for the values of ω are summarized in Table 3.1. Stability with respect to a reasonable degree of scaling is demonstrated by the examples shown in Figures 3.39 and 3.40, where the locations of the landmarks obtained for the specialty plier

which has been scaled by an area factor of 0.5 and 2.1, respectively, are relatively invariant. One drawback of this approach is that small desirable details may be deleted by the smoothing. On the other hand, an algorithm that can detect small details may at the same time introduce many false details.

3.5. Summary

Two widely used polygonal approximation algorithms along with their drawbacks have been discussed. We have developed two new methods to detect landmarks from contours. The first method is known as the curvature guided polygonal approximation. It is based on the fact that break points resulting from a polygonal approximation of an object boundary are mostly extreme curvature points of the boundary. Smoothing is carried out to avoid excessive false concavities and convexities. A more robust approach is introduced that uses the cardinal curvature points of an object boundary as the landmarks. The number of landmarks obtained using this approach is usually smaller than that using the curvature guided polygonal approximation; hence, less computation is required for the higher level landmark matching processing stage.

Table 3.1.
Range of ω values used to obtain the landmarks
of various objects based on cardinal curvature points

Models	Figures	Range of ω
wrench	3.35	14.5-24
needle-nose plier	3.36	14.5-(>100)
wire cutter	3.34	16-28.5
specialty plier	3.37	14-40.5
wire stripper	3.38	16.5-36.5

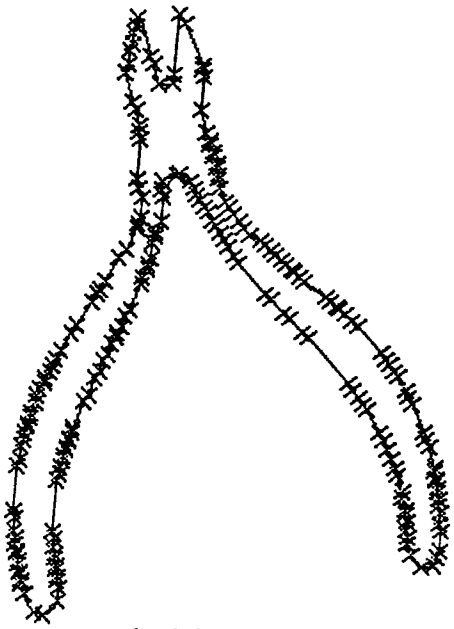


Figure 3.18. The Gaussian smoothed boundary of the wire cutter using ω=1. Each "×" indicates an extreme curvature point.

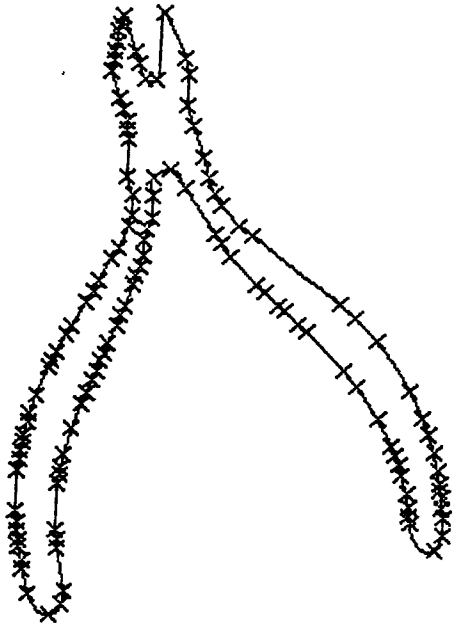


Figure 3.19. The Gaussian smoothed boundary of the wire cutter using ω=2. Each "×" indicates an extreme curvature point.

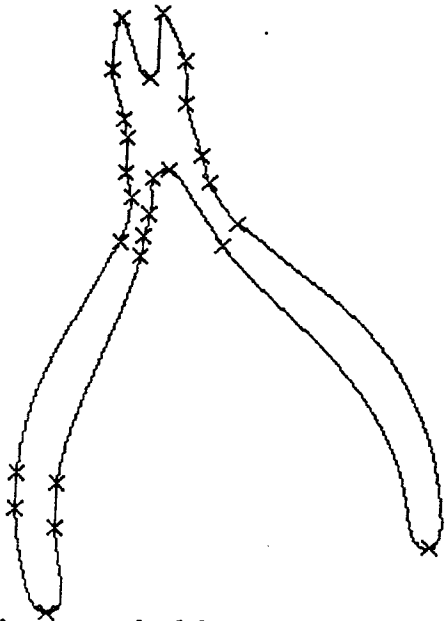


Figure 3.20. The Gaussian smoothed boundary of the wire cutter using ω=4. Each "X" indicates an extreme curvature point.

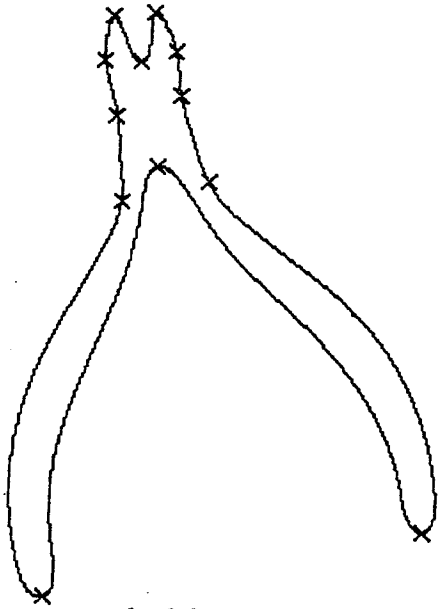


Figure 3.21. The Gaussian smoothed boundary of the wire cutter using ω=8. Each "X" indicates an extreme curvature point.

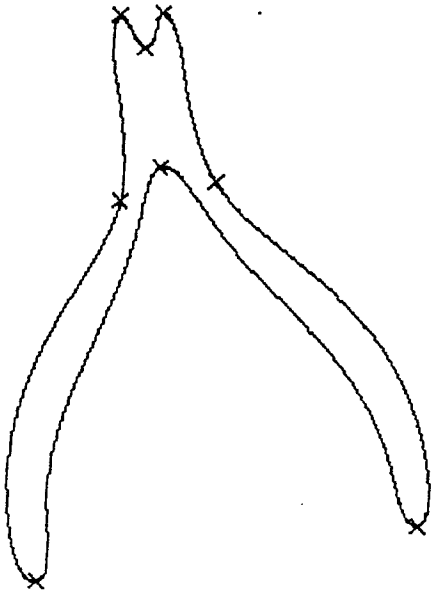


Figure 3.22. The Gaussian smoothed boundary of the wire cutter using ω=12. Each "×" indicates an extreme curvature point.

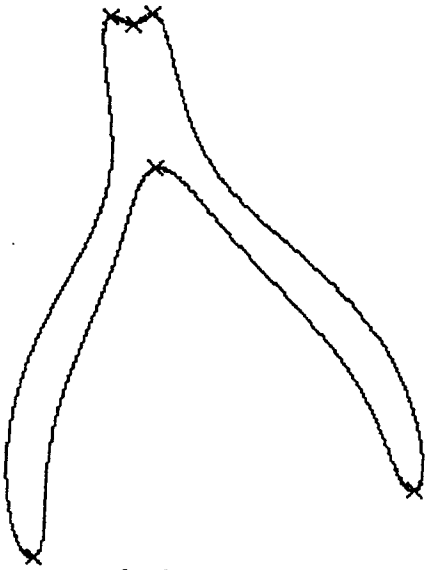


Figure 3.23. The Gaussian smoothed boundary of the wire cutter using ω=20. Each "×" indicates an extreme curvature point.

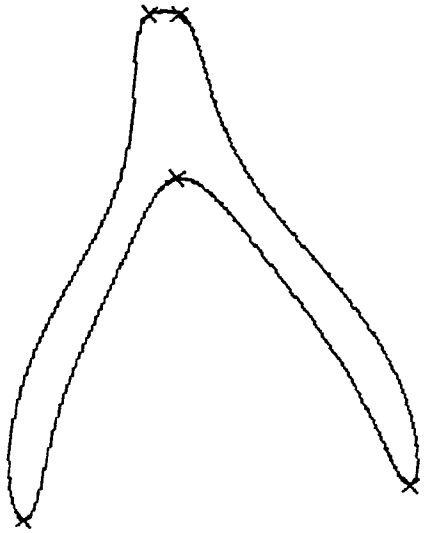


Figure 3.24. The Gaussian smoothed boundary of the wire cutter using ω=30. Each "×" indicates an extreme curvature point.

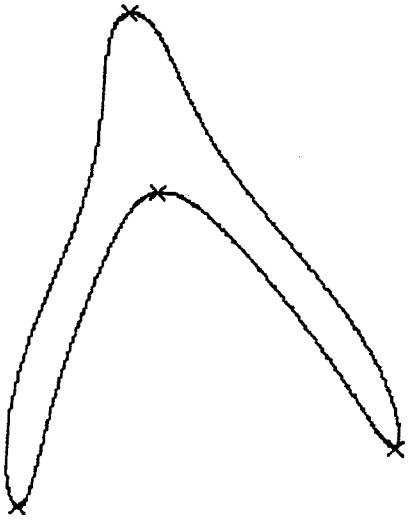


Figure 3.25. The Gaussian smoothed boundary of the wire cutter using ω=40. Each "×" indicates an extreme curvature point.

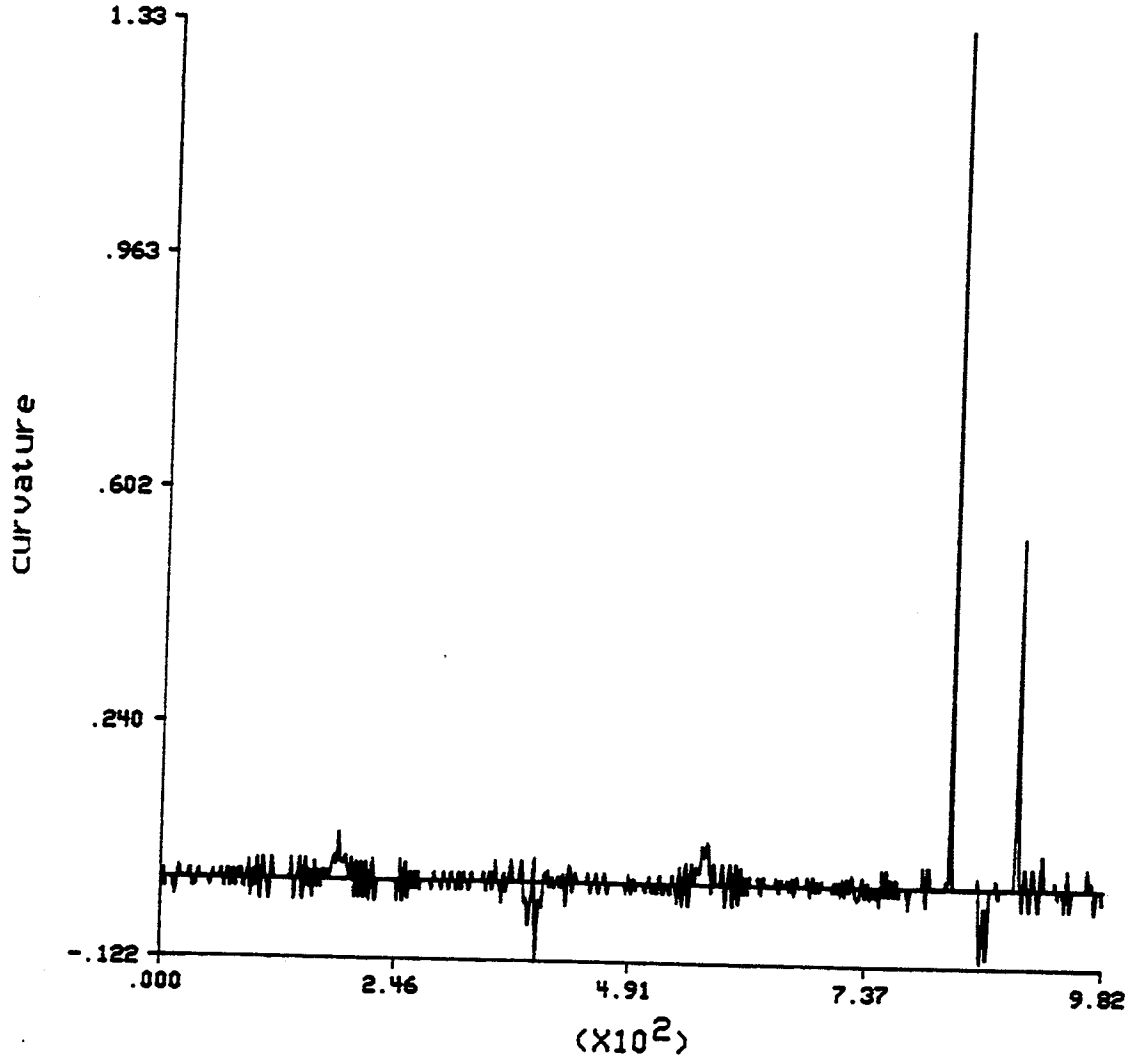


Figure 3.26. The curvature function of the Gaussian smoothed boundary of the wire cutter using ω=1.

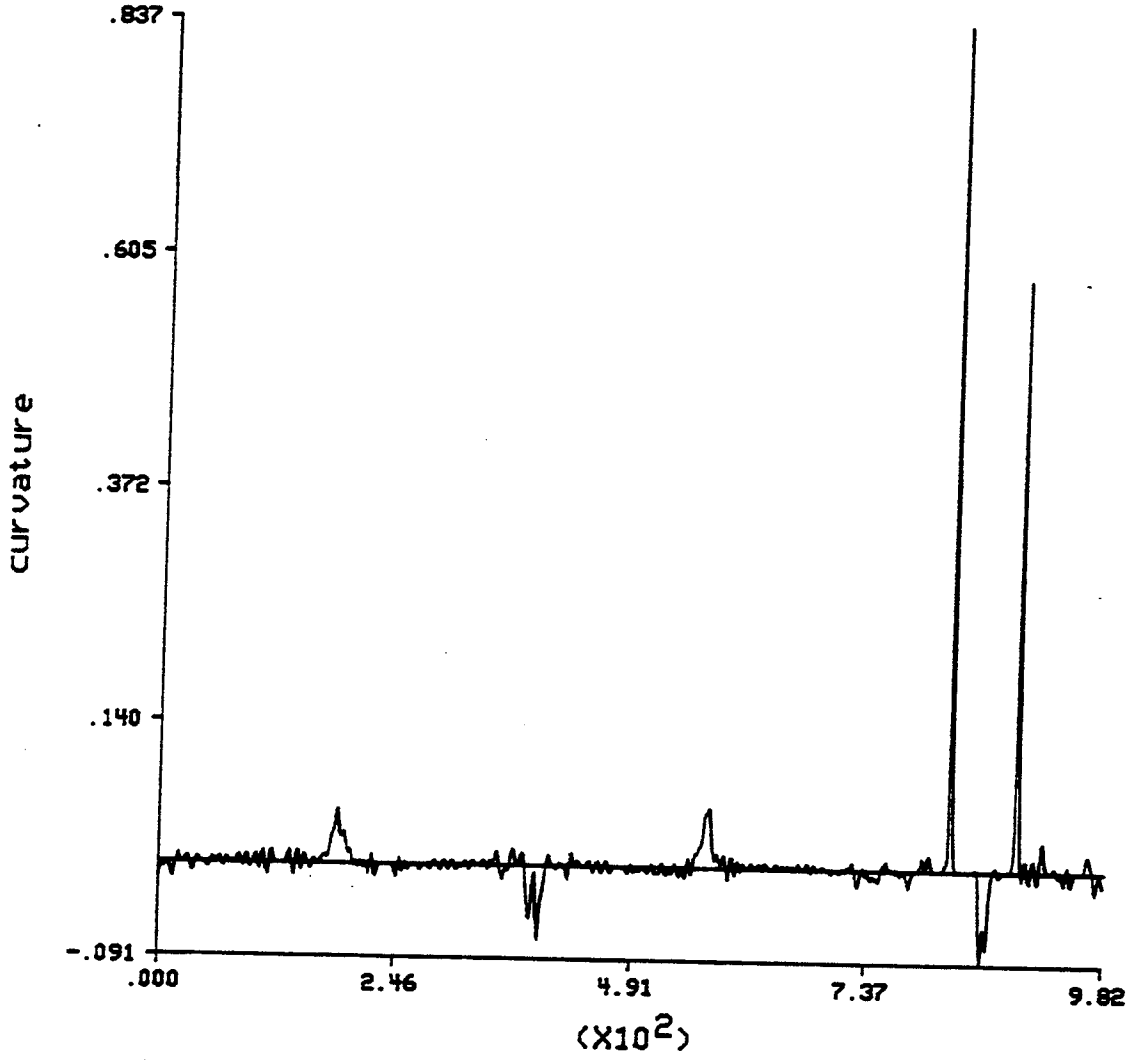


Figure 3.27. The curvature function of the Gaussian smoothed boundary of the wire cutter using ω=2.

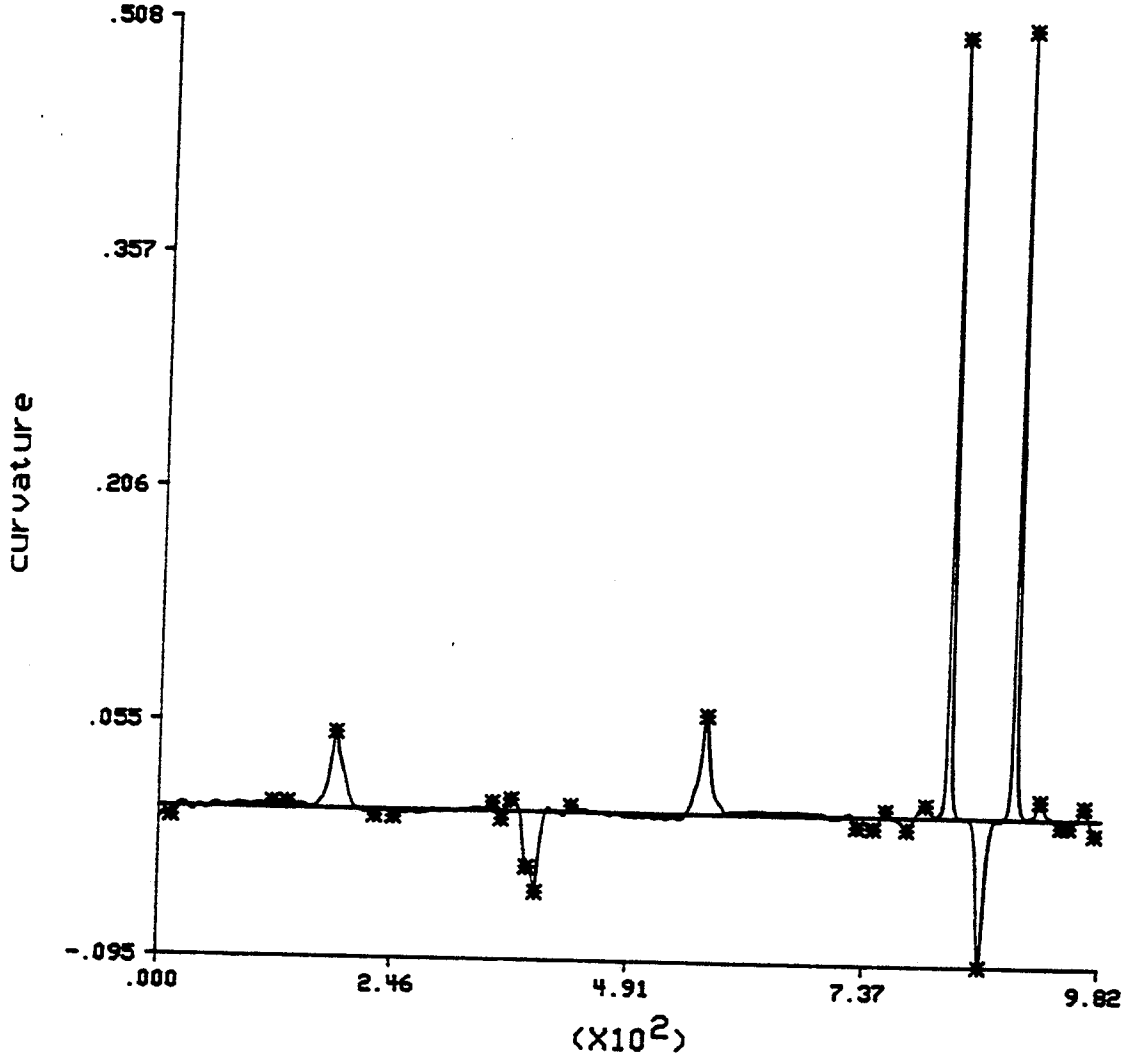


Figure 3.28. The curvature function of the Gaussian smoothed boundary of the wire cutter using ω=4. Each "*" indicates an extreme curvature point.

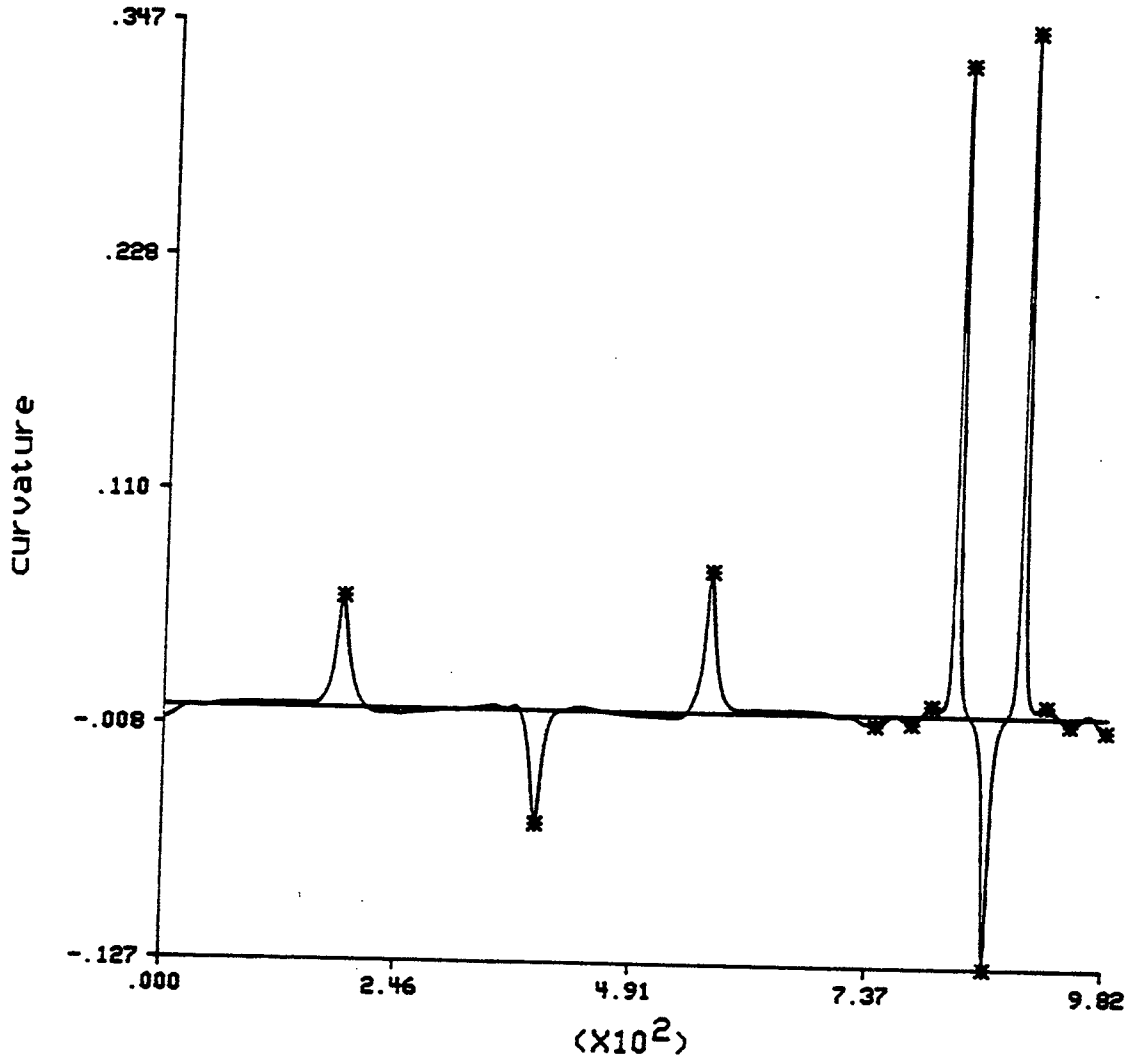


Figure 3.29. The curvature function of the Gaussian smoothed boundary of the wire cutter using ω=8. Each "*" indicates an extreme curvature point.

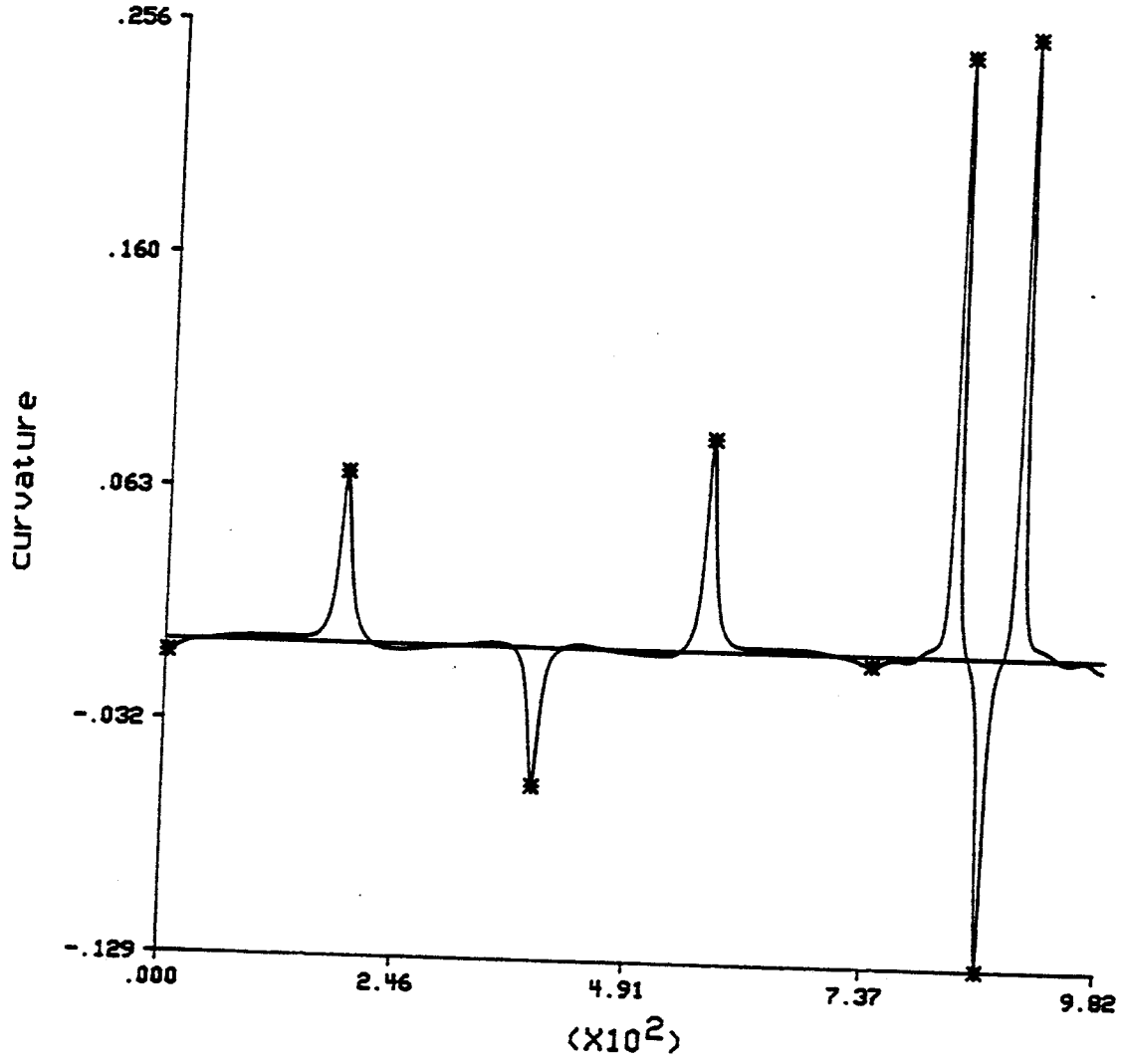


Figure 3.30. The curvature function of the Gaussian smoothed boundary of the wire cutter using ω=12. Each "*" indicates an extreme curvature point.

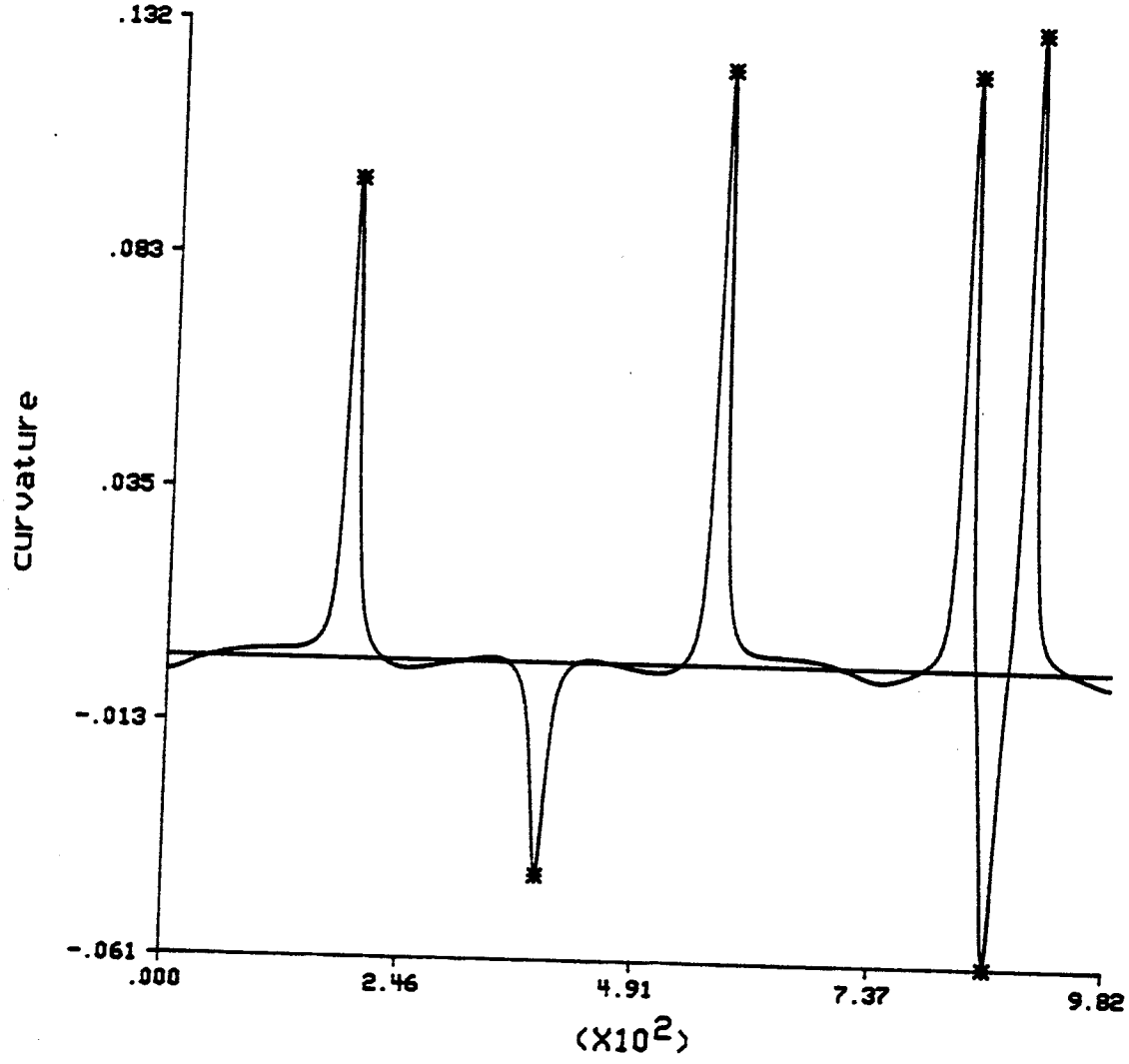


Figure 3.31. The curvature function of the Gaussian smoothed boundary of the wire cutter using ω=20. Each "*" indicates an extreme curvature point.

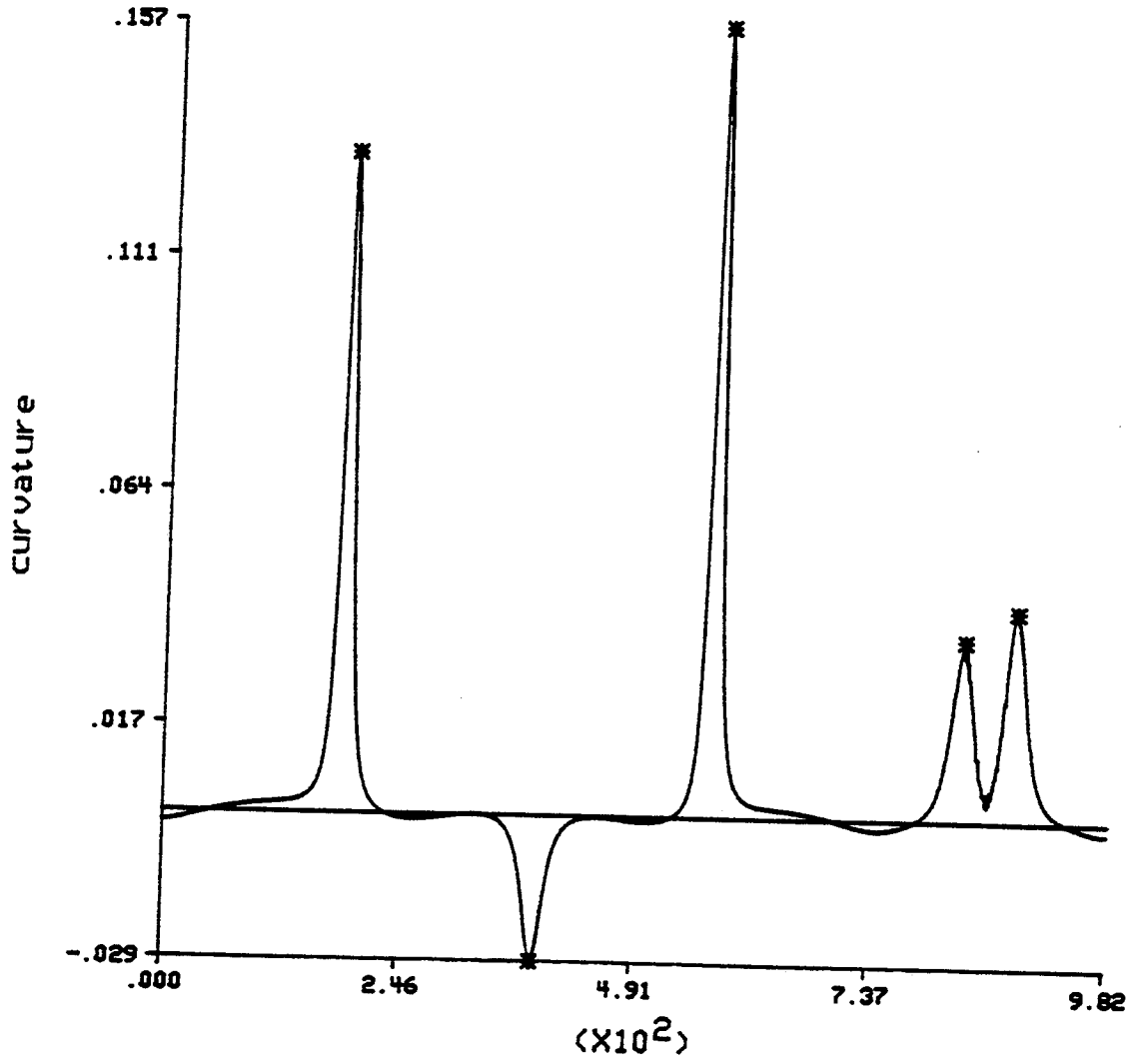


Figure 3.32. The curvature function of the Gaussian smoothed boundary of the wire cutter using ω=30. Each "*" indicates an extreme curvature point.

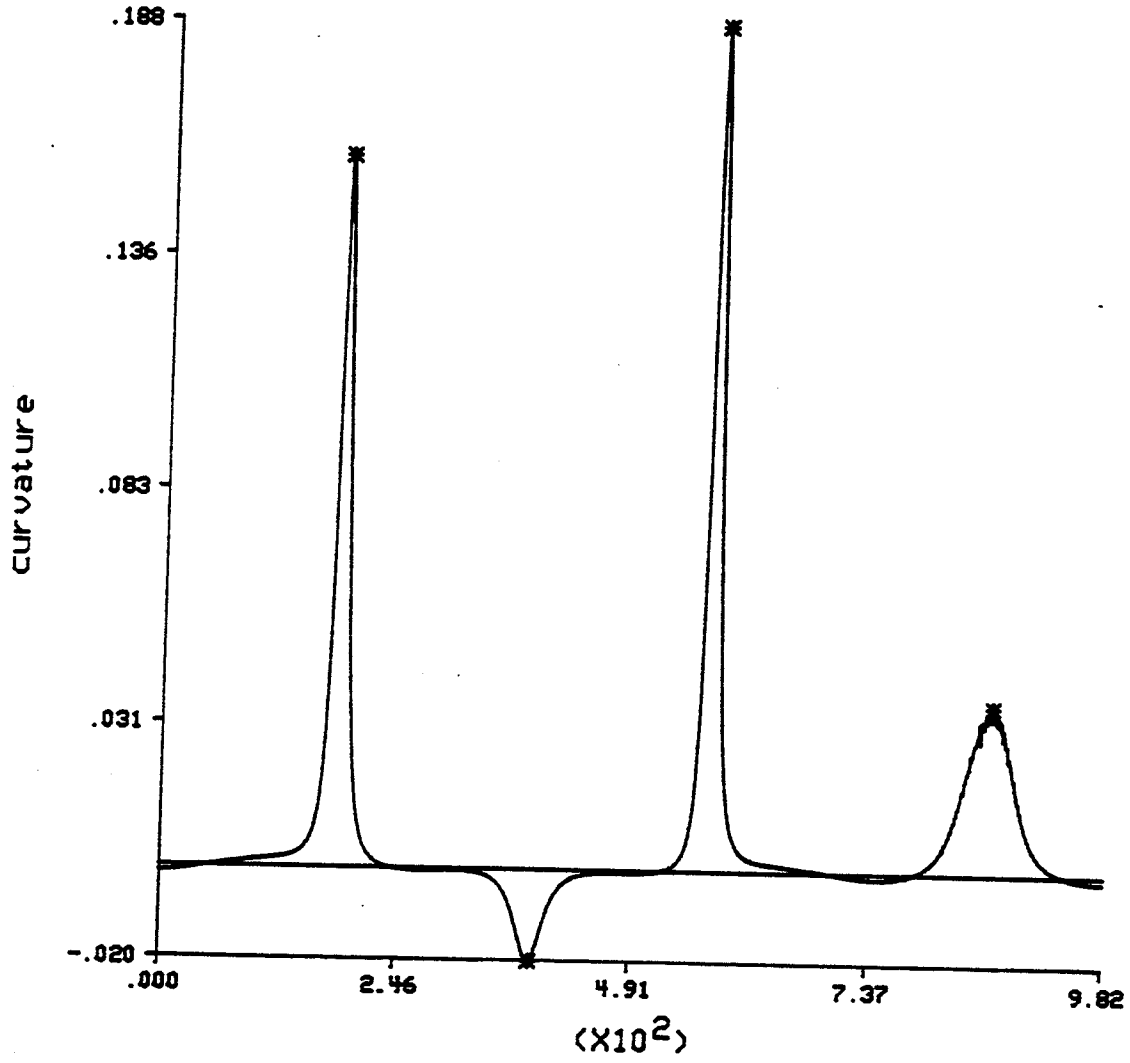


Figure 3.33. The curvature function of the Gaussian smoothed boundary of the wire cutter using ω=40. Each "*" indicates an extreme curvature point.

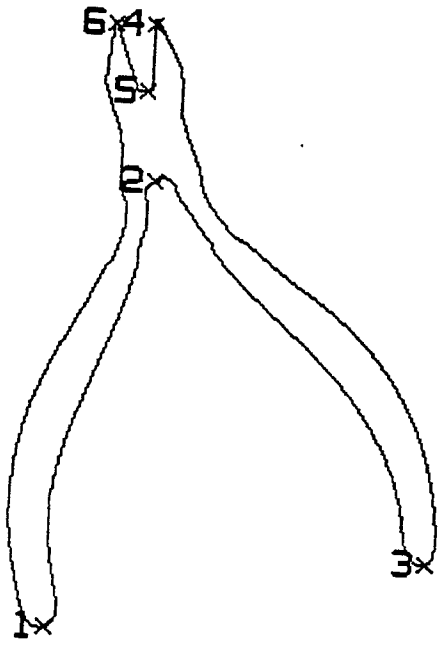


Figure 3.34. The landmarks of the wire cutter based on the cardinal curvature points. Each landmark is numerically labeled, and is indicated by an "×."

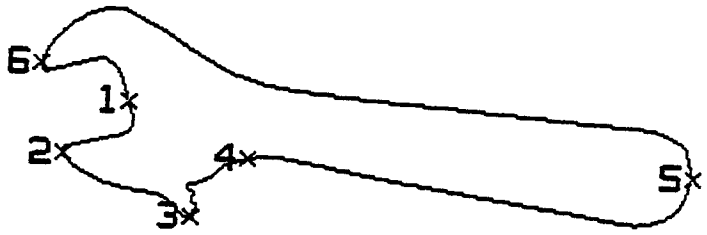


Figure 3.35. The landmarks of the wrench based on the cardinal curvature points. Each landmark is numerically labeled, and is indicated by an "×."

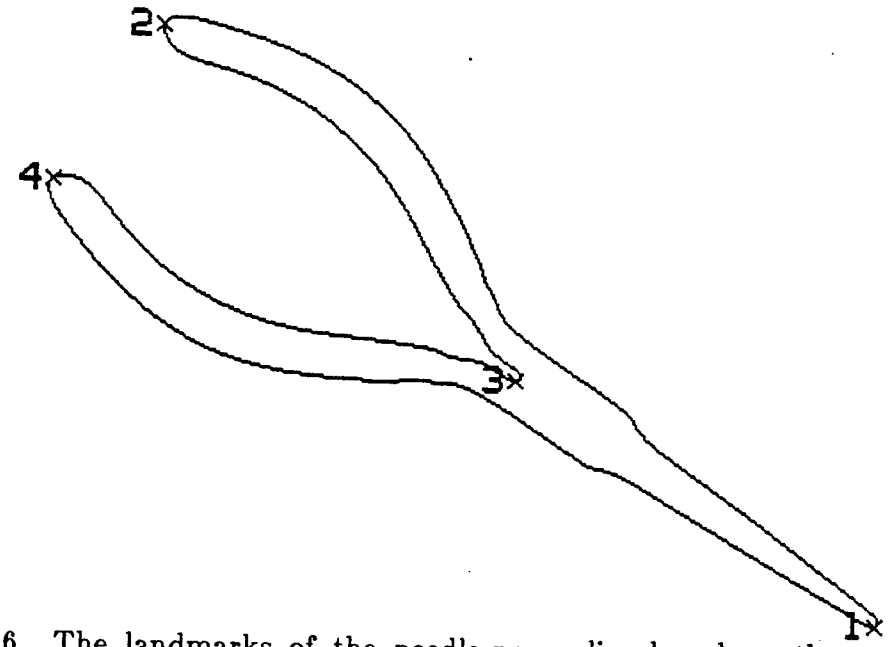


Figure 3.36. The landmarks of the needle-nose plier based on the cardinal curvature points. Each landmark is numerically labeled, and is indicated by an "×."

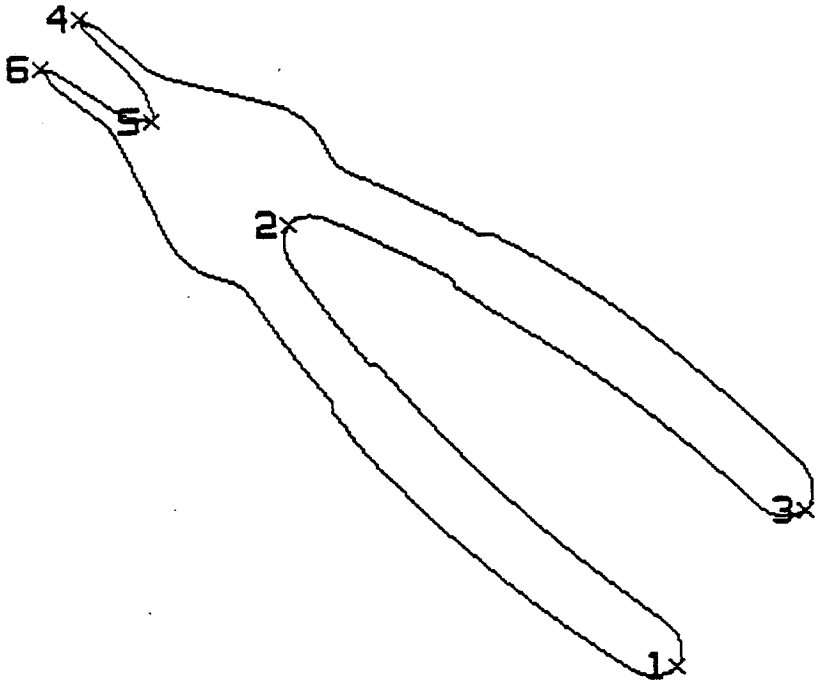


Figure 3.37. The landmarks of the specialty plier based on the cardinal curvature points. Each landmark is numerically labeled, and is indicated by an "×."

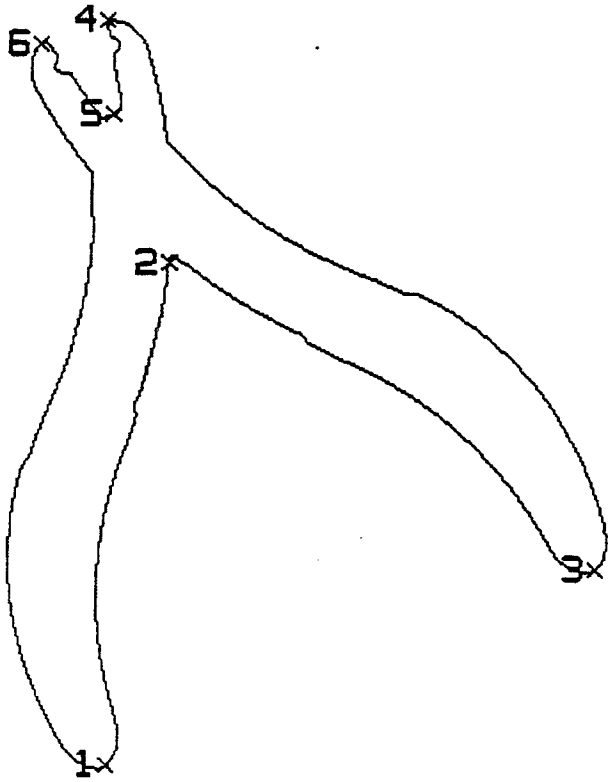


Figure 3.38. The landmarks of the wire stripper based on the cardinal curvature points. Each landmark is numerically labeled, and is indicated by an "×."

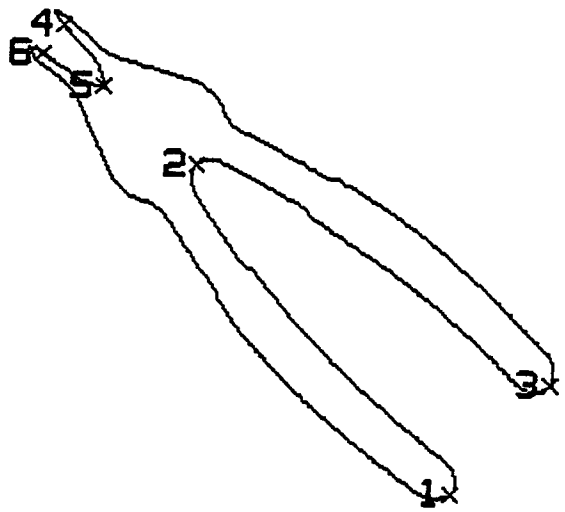


Figure 3.39. The landmarks of the specialty plier, scaled by an area factor of 0.5, based on the cardinal curvature points. Each landmark is numerically labeled, and is indicated by an "×."

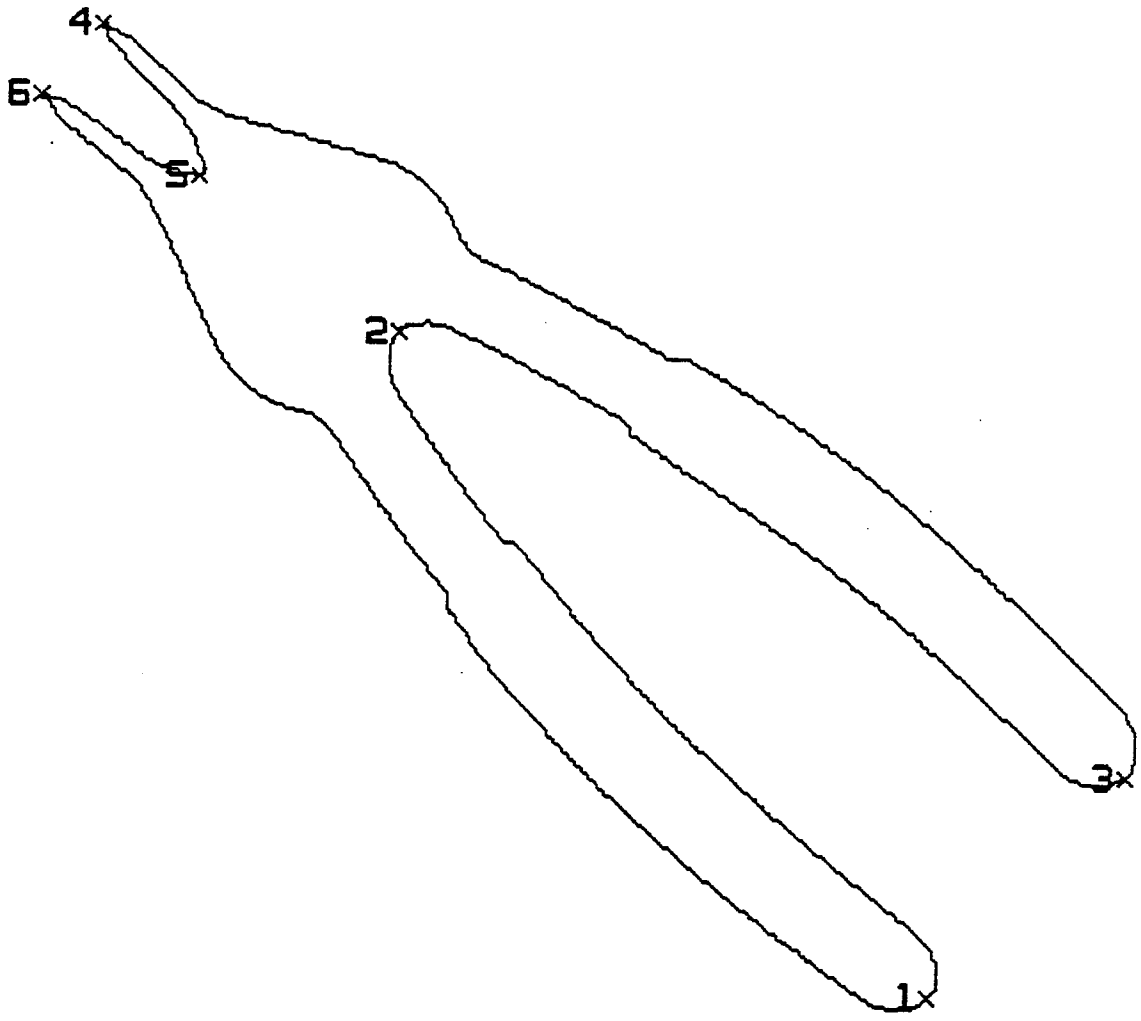


Figure 3.40. The landmarks of the specialty plier, scaled by an area factor of 2.1, based on the cardinal curvature points. Each landmark is numerically labeled, and is indicated by an "×."

CHAPTER 4

LANDMARK MATCHING AND LOCATION ESTIMATION

4.1. Introduction

In the previous chapters, we have developed two methods to extract landmarks. We have also shown that sphericity is a robust shape measure. In this chapter, we shall describe the landmark matching task and the decision strategy of our landmark-based shape recognition approach (see Figure 1.4). Before we discuss the landmark matching task, we shall first review recent work on 2-D partial shape recognition in Section 4.2. Landmark matching, location estimation, and matching verification will be discussed in Section 4.3. The computational complexity of the landmark matching task will be evaluated. Some experimental results will be presented in Section 4.4. A summary of this chapter will then be given in Section 4.5.

4.2. Literature Review

Recent work on 2-D partial shape recognition has exhibited an increasing interest in developing methods capable of recognizing objects when global information about the objects are not available. We shall discuss several methods reported in the recent literature.

Bolles and Cain [Bol82] use a hypothesis generation and verification approach to recognize and locate partially visible objects. The shape features of an object are holes and corners (right-angled corners are used in the paper). The physical description of the features, such as the size of a hole and the included angle of a corner, are used to indicate the similarity between a model feature and a scene feature. The structural relationships among the shape features are then exploited to construct a structured graph. A node of the graph corresponds to an assignment pair indicating a possible match between a model and a scene feature based on their physical description. Two nodes that are mutually compatible in structure and meet certain criteria are connected with an arc. Two nodes are said to be structurally compatible if the physical

distance and the relative orientation between two scene features are within certain limits with respect to those of the two model features. The hypothesis of a model in a scene is generated by finding the largest completely connected subgraph using an algorithm known as a *maximal-clique* algorithm. The largest connected subgraph corresponds to the largest set of structurally compatible matches of the graph. The location and orientation of the model in the scene is estimated from the matched pairs of the largest connected subgraph. The model is then translated and rotated onto the scene. The hypothesis is then verified by checking the boundary consistency between the scene and the coordinate transformed model. Since the goodness of match between the model features and the scene features is determined by the physical and structural descriptions of the features which are not scale invariant, this approach is susceptible to scale variations. In addition, the clique finding algorithm is very complex and computationally intensive.

Bhanu and Faugeras [Bha84] cast the shape matching problem as a segment matching problem. An object contour is first approximated by a polygon from which feature values 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).

This approach exemplifies an application of the relaxation labeling method in the computer vision area. It is computationally intensive. A good estimate of the initial assignment of the label is important to the convergence of the approach and the validity of the result. In addition, feature values such as the length of a segment and the intervertice distance are scale dependent, the shape measure based on these feature values are thus sensitive to scale variations. Therefore, the algorithm cannot recognize objects in a scene that have a different scale from that of the models.

A simple technique to solve the occlusion problem has been proposed by Price [Pri84]. The shape features of an object are the line segments of the approximated polygon of the object. Each model segment is then compared with every scene segment in terms of their lengths, and the included angles between successive segments. If the lengths and the angles 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 matching 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 considered for the starting location of the longest sequence. Furthermore, the technique is sensitive to scale variations because the feature value, such as the length of a line segment, used in this technique is inherently scale dependent.

Bhanu and Ming [Bha87] improve upon 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 a 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 matching to the total length of all segments of the model is evaluated to verify the final matching.

The approach is capable of recognizing occluded objects. However, it suffers the same sensitivity to scale variations. Though it is computationally more efficient than Price's approach, it remains computationally expensive because of the iterative nature of the algorithm.

Ayache and Faugeras [Aya86] develop a method known as HYPER (HYpotheses Predicted and Evaluated Recursively) to recognize and position 2-D objects. The shape features of an object are the line segments of the approximated polygon of the object. The longest model segments are called the "privileged" segments. A hypothesis is made by matching a "privileged" model segment with a scene segment based on some compatibility criteria. The two

segments are compatible if (1) the included angles of the segments made with their respective preceding segments are approximately equal to each other, and (2) the ratio between the lengths of the two segments is close to an *a priori* estimated scale factor. Such criteria will usually generate many hypotheses. The number of hypotheses is reduced by limiting the number of "privileged" model segments and selecting only the best few hypotheses for further processing. A coordinate transformation including translation, rotation, and scaling taking the "privileged" model segment onto the scene segment is estimated. Based on a matched pair between a "privileged" model segment and a scene segment, additional segments between the model and the scene having a small dissimilarity measure are matched. The dissimilarity measure between a model and a scene segment is a weighted sum of the differences of the orientations, lengths, and the Euclidean distance between the two segments. For each additional matched pair, the coordinate transformation is updated by a Kalman filter, and a quality measure which accounts for the relative length of the model segments that have been identified is computed. The matching process ends when a large enough number of hypotheses have been evaluated, or when a very high quality measure of a hypothesis is reached. The hypothesis having the highest quality measure is finally reexamined using the last estimated parameters of the coordinate transformation as the initial estimation. Using these parameters as the initial estimates of the coordinate transformation, the process is repeated until it converges. The reexamined hypothesis is finally validated or rejected based on its quality measure.

The shape measures (the compatibility measure and the dissimilarity measure) used in this approach are not unique; i.e., segments that are locally different could yield similar shape measures. Unless an estimate of the scale factor based on *a priori* knowledge is provided, the approach is sensitive to scale variations. The iterative nature of this approach also makes it computationally expensive.

Koch and Kashyap [Koc87] use a hypothesis generation and verification approach, a concept similar to [Bol82], to solve the partial recognition problem. However, the shape features, the shape measures, and the matching method are different from those of [Bol82]. Each object is first approximated by a polygon from which corner points are extracted. According to [Koc87], a *corner* defines a group of line segments centered at a corner vertex. To match two corners, the polygon fragments associated with each corner are first quantized into a same number of points. Then a coordinate transformation consisting of

rotation and translation that minimizes the squared error between the points of the polygon fragments is determined. The resulting minimum squared error is the shape measure used to indicate the dissimilarity between the two corners. It involves the first-order, second-order, and cross moments of the polygon fragments of the two corners. These moments can however be computed by using the vertices of the polygon fragments, and thus a large amount of computation is reduced.

The hypothesis generation involves two steps. The first step determines the compatibility of the matches between the model and the scene corners. Matches that are considered compatible must satisfy certain constraints. The final hypothesis is reached by growing a cluster around a good match. The resulting cluster is a group of matches between the model and the scene corners. The least squared coordinate transformation that transforms a group of model corner points onto a group of scene corner points in a least squares sense is also obtained. To verify the hypothesis that the model is in the scene, the hypothesized model polygon is applied with the above least squared coordinate transformation. The match error of the hypothesis is the difference between the area enclosed by the scene polygon and the area enclosed by the scene polygon unioned with the transformed model polygon. A large match error results in the rejection of the hypothesis while a small one results in the acceptance of the hypothesis. The algorithm has assumed that the model and the scene are of the same scale. It is sensitive to scale variations because the least squared error between a model and a scene polygon will depend on the scale of both the model and the scene polygons.

Turney *et al.* [Tur85] employ a template matching algorithm to recognize partially occluded objects. The template of an object is its boundary. It is subdivided into subtemplates which are portions of the object boundary. Each subtemplate is associated with a value known as its significant value which indicates the importance of the subtemplate. A subtemplate having a high significant value is considered as a distinctive feature. Matching is done by cross-correlating each subtemplate with the scene boundary in the angle versus arc length space. The angle at a boundary pixel is the angle of the tangent at that pixel. A matching coefficient which indicates the goodness of match between a scene boundary segment and a subtemplate is computed. This coefficient is weighted by the significant value of the subtemplate.

This algorithm can recognize a partially occluded object provided that the distinctive boundary segments associated with the object is not occluded. A

training phase is required to determine the weight of each subtemplate of an object template. Again, matching in the angle versus arc length space is sensitive to scale variations.

In contrast to Turney *et al.*'s use of distinctive features, Knoll and Jain [Kno86] emphasize features that are common to several objects to hypothesize for object identities and orientations. A list is associated with each feature that indicates where it occurs in each model. When a match of a feature with the scene is found, models having such a feature are hypothesized for their identities and orientations from the feature's list. Each of these hypotheses is then tested and verified. Knoll and Jain call this approach the feature indexed hypotheses method.

The features used are fixed length boundary segments of an object. The shape measure between two features is the sum of the point-wise Euclidean distances between the two appropriately aligned boundary segments. The hypothesis test involves a variation of template matching and the use of several heuristics. A score is generated for each hypothesis test. A negative score indicates a negative evidence of the hypothesis. The hypothesis having the highest score is the most confident hypothesis.

This approach alleviates the problem of having distinctive features occluded. If the number of matches per feature can be controlled, the recognition time can be made proportional to the square root of the size of the model set. However, this number is usually difficult to determine especially when objects are occluded in which case features may disappear or multiply. Since the features are fixed length boundary segments and their number is restricted, this approach is also sensitive to scale variations.

Gorman and Mitchell [Gor88] represent an object contour by breaking the contour into contour segments. The break points 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 inter-segment distance table measuring the norm squared distances between the model and the scene contour segments is constructed. The row index of the table indicates a model segment, and the

column index indicates a scene segment. Entry (i, j) is referred to as the ith row and the jth column entry of the table. It stores the norm squared distance between the Fourier coefficients of the ith model segment and the jth scene segment. The table is augmented by repeating the rows.

A backward dynamic programming procedure is 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. They use a criterion for path completeness requiring that the path must make use of all scene segments. This means that the path must traverse through every column of the table, from the first column to the last column of the table. This criterion seems inadequate for two reasons. First, the scene may consist of more than one object overlapping each other, and hence has more segments than the model. Second, the scene may also have only one object being occluded and have less segments than the model. Therefore, the path should not necessarily make use of all scene segments. In addition, if the first segment of the scene contour does not match with any segment of the model, the minimum distance path may be swayed from the path of true matches resulting in false matches. However, this approach is not sensitive to scale variations because the Fourier coefficients have been normalized.

4.3. Landmark-Based Shape Recognition — Landmark Matching, Location Estimation, and Matching Verification

Our shape recognition algorithm is based on an approach that is completely different from the above methods. We use different shape features, a different shape measure, and a different feature matching algorithm. Our shape features of an object are the landmarks associated with the object. In Chapter 3, we have presented two methods of extracting landmarks. Instead of evaluating many feature values in order to characterize the similarity between two line segments, we use sphericity to discriminate the dissimilarity between two landmarks. Sphericity has been discussed in detail in Chapter 2, and has been shown to be translation, rotation, and scale invariant. It is also relatively robust with respect to distortion. In contrast to all the above methods except [Gor88], our approach is not sensitive to scale variations. Our feature matching algorithm is not iterative. We use an algorithm which we call HOPPING dynamic programming which switches between a forward and a backward dynamic programming procedure to perform the landmark matching

task. Before we describe our matching procedure, we shall first discuss properties of dynamic programming in the next section.

4.3.1. Dynamic Programming

Dynamic programming has found applications in many areas. It is the study of multistage decision processes. Dreyfus and Law [Dre77] have concisely described *dynamic programming* as follows:

> "Dynamic programming is an optimization procedure that is particularly applicable to problems requiring a sequence of interrelated decisions. Each decision transforms the current situation into a new situation. A sequence of decisions, which in turn yields a sequence of situations, is sought that maximizes (or minimizes) some measure of value. The value of a sequence of decisions is generally equal to the sum of the values of the individual decisions and situations in the sequence."

We shall illustrate the concept of a multistage decision process with a simple classical example. We consider a path problem shown in Figure 4.1a. Letters $\{A, B, \cdots, I\}$ denote the names of the cities. The number along the line joining two cities indicates the distance between the two cities. The problem is to find the shortest path traveling from city A to city I. To travel from city A to city I, we have to pass through several intermediate cities. Each of these intermediate cities can be thought of as the state of the overall process at an intermediate stage. Since there are more than one city that is reachable from a given city, a decision of which city to reach has to be made at each stage. Consequently, this process is called a *multistage decision process.* We need a *policy* of making decisions at each stage so as to achieve the shortest path between city A and city I. Such a policy must satisfy the *principle of optimality* [Bel65]:

> "An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision."

The above path problem can be mathematically formulated as follows:

Let $S = \{A, B, \cdots, I\}$ be the state space,

$s_i \in S$ be the state at which the process is at the ith stage,

$s_0 = A$ be the state at the initial stage,

$s_n = I$ be the state at the final stage.

We want to determine the multistage decision process $\{s_0, s_1, \cdots, s_n\}$ such that the total distance from stage s_0 to s_n is minimum.

Denote R_D as the minimum distance path from D to I, and
$d(A, D)$ as the distance between A and D.
Using the principle of optimality, we reach the following recursive solution for the shortest path problem:

$$R_A = \min \begin{cases} d(A, D)+R_D \\ d(A, B)+R_B \end{cases} ,$$

$$R_D = \min \begin{cases} d(D, E)+R_E \\ d(D, G)+R_G \end{cases} , \quad R_B = \min \begin{cases} d(B, C)+R_C \\ d(B, E)+R_E \end{cases} ,$$

$$\vdots$$

$$R_F = d(F, I) \quad , \quad R_H = d(H, I) \ .$$

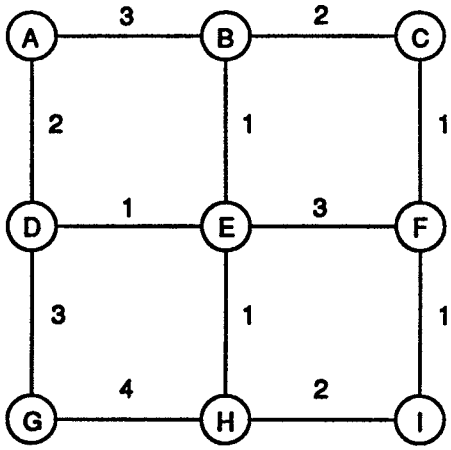
The above formulation is known as a *backward* dynamic programming procedure [Dre77] since the multistage decision process is determined by working backward from the destination point to the starting point. The shortest path using this formulation is shown in Figure 4.1b. The feature matching procedure of [Gor88] is such a procedure where the starting point and the destination point can be any point in the first and the last column, respectively, of their augmented inter-segment distance table.

We can paraphrase the reverse version of the principle of optimality as follows:
"An optimal policy has the property that whatever the final state and final decision are, the decisions that have been made so far with regard to the state prior to the final decision must constitute an optimal policy."
A different formulation of the above path problem can be obtained using the reverse version of the principle of optimality. Using the same notations as above except that R_D now denotes the shortest distance path from A to D, we reach another recursive solution to the problem:

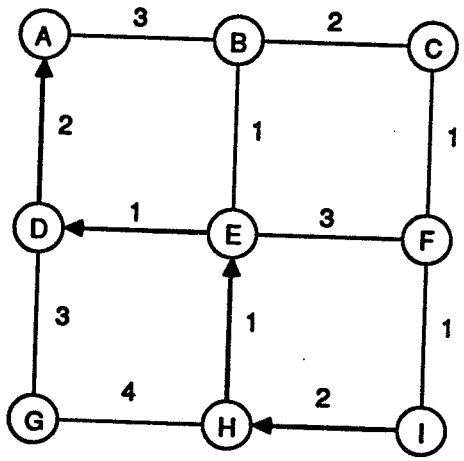
$$R_I = \min \begin{cases} d(F, I)+R_F \\ d(H, I)+R_H \end{cases} ,$$

$$R_F = \min \begin{cases} d(C, F)+R_C \\ d(E, F)+R_E \end{cases} , \quad R_H = \min \begin{cases} d(E, H)+R_E \\ d(G, H)+R_G \end{cases} ,$$

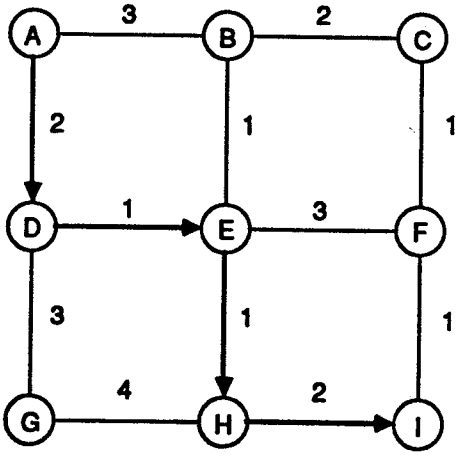
$$\vdots$$



(a)



(b)



(c)

Figure 4.1. An example of a classical shortest path problem. (a) A shortest path problem. (b) The shortest path shown by arrows results from a backward dynamic programming procedure. (c) The shortest path shown by arrows results from a forward dynamic programming procedure.

$$R_B = d(A, B) \quad , \quad R_D = d(A, D) \ .$$

This formulation is known as a *forward* dynamic programming procedure [Dre77] since the multistage decision process is determined by working forward from the starting point to the destination point. The shortest distance path using this formulation is shown in Figure 4.1c.

In general, each formulation yields a different solution. The backward dynamic programming is usually applied when only the destination point is available, while the forward dynamic programming is applied when only the starting point is available. When both the starting and the destination point are known, either backward or forward procedure can be applied.

4.3.2. Landmark Matching by Hopping Dynamic Programming

Our problem of matching landmarks of a model to those of a scene is equivalent to that of matching two sequences of landmarks associated with the model and the scene.

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 the coordinates of a sequence of landmarks associated with a scene.

Note that n is the number of model landmarks, and m is the number of scene landmarks. The subscripts denote the order of the landmarks. The goodness of match between the ith model landmark and the jth scene landmark is given by the sphericity (Equation 2.13) derived from a triangular transformation mapping $\{(x_{i-1}, y_{i-1}), (x_i, y_i), (x_{i+1}, y_{i+1})\}$ to $\{(u_{j-1}, v_{j-1}), (u_j, v_j), (u_{j+1}, v_{j+1})\}$. At the end points, when i=1, $i-1$ is replaced by n; when j=1, $j-1$ is replaced by m; when i=n, $i+1$ is replaced by 0; when i=m, $i+1$ is replaced by 0. These replacements are to account for the periodic arrangement of the landmarks. A mapping is said to be ***orientation or sense reversing*** [O'N66] if the Jacobian of the mapping is negative. To account for the sense of a mapping, we negate the value of the sphericity if the triangular transformation is sense reversing. Thus, the sphericity derived from mapping the ith model landmark to the jth scene landmark having a value close to 1 implies that these two landmarks are locally similar.

A table of compatibility is constructed between the sequence of model landmarks and the sequence of scene landmarks. The row index indicates a model landmark while the column index indicates a scene landmark. Entry

(i, j) is referred to as the ith row and the jth column entry of the table. 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 respective adjacent landmarks. Consider a simple example of a scene where there are two objects overlapping each other as shown in Figure 4.2. The extracted landmarks in the scene are based on the cardinal curvature points using ω=20. A table of compatibility between the wire stripper (Figure 3.38) and the scene (Figure 4.2) is shown in Figure 4.3a. 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 of finding such a sequence is impractical. We will instead formulate a dynamic programming procedure to achieve this matching.

Our matching procedure is slightly similar to the feature matching algorithm of [Gor88]. As mentioned earlier, Gorman and Mitchell [Gor88] use a backward dynamic programming procedure to find a minimum distance path from the first column to the last column of their augmented inter-segment 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 shall 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. Unlike the shortest path problem, neither the starting point nor the destination point of a path which corresponds to a sequence of matches between the scene and model landmarks are known. It is not *a priori* known how many landmarks of a model will match with those of a scene. Instead of having a starting and a destination point, a ***support entry***, which is an entry in the table that provides strong evidence of a true match between a model and a scene landmark, is used to guide the matching process. This evidence is strong if the entry as well as its diagonal neighboring entries have sphericity values close to 1. 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

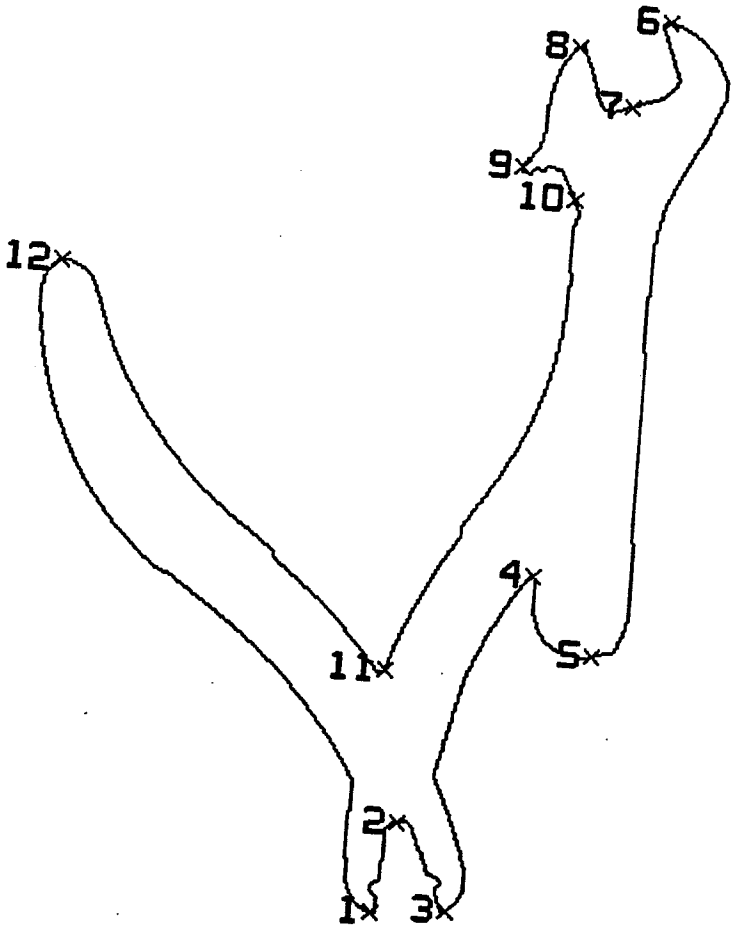


Figure 4.2. A scene which consists of a wire stripper and a wrench overlapping each other. Each scene landmark is labeled and indicated by an "X."

model landmarks	1	2	3	4	5	6	7	8	9	10	11	12
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

scene landmarks

(a)

model landmarks	10	11	12	1	2	3
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

scene landmarks

(b)

model landmarks	10	11	12	1	2	3
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

scene landmarks

(c)

Figure 4.3. An example of performing the landmark matching task between the wire stripper and the scene shown in Figure 4.2. (a) The table of compatibility. (b) The result of performing hopping dynamic programming using (3, 12) as the support entry. (c) The resulting path indicated by 1's is the maximum value path.

Figure 4.3a, 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 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 with 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.

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

By 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 backward or forward dynamic programming, 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 the backward procedure is applicable when the destination point is available, and the forward procedure is applicable when the starting point is available, the support entry can be treated both as 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 (k, l) to (i, j) entry in the backward procedure, and

$a_f(i, j)$ as the accumulated sum of the sphericity values from (k, l) to (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 and the $(j-1)$th scene landmark, and hence the sphericity value at $(i-1,j-1)$ is added to the accumulated sum of sphericity values at (i, j) to produce the accumulated sum of sphericity value at $(i-1,j-1)$. Since a horizontal or a vertical transition does not constitute a match, the accumulated sum of sphericity values 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 we are matching, 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 and the $(j+1)$th scene landmark, and hence the accumulated sum of sphericity values at $(i+1, j+1)$ is obtained by the sum of the sphericity value at $(i+1, j+1)$ and the accumulated sum of sphericity value at (i,j). Likewise, Rules (2) and (3) are the boundary conditions, and Rule (4) is the initial condition. To account for the periodic nature of the landmarks we are matching, 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 $j+1-m$.

How do we switch between the forward and the 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 $(\tilde{i}, \tilde{j})$ entry be where the backward procedure has reached at the present stage, and $(\hat{i}, \hat{j})$ entry be where the forward procedure has reached at the present stage. We define the backward average sphericity value at entry $(\tilde{i}, \tilde{j})$ as $a_b(\tilde{i}, \tilde{j})$ divided by the number of transitions made by the backward procedure traversing from entry (k, l) to entry $(\tilde{i}, \tilde{j})$ 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 $(\tilde{i}, \tilde{j})$ is larger than the forward average sphericity value at entry $(\hat{i}, \hat{j})$, the backward procedure will proceed to entry $(\tilde{i}-1, \tilde{j}-1)$; otherwise, the forward procedure will proceed to $(\hat{i}+1, \hat{j}+1)$. In other words, the procedure that has 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 called the *maximum value path.* Because of the periodic nature of the landmarks we are matching, either path can wrap around the table. We call this hopping dynamic programming (*HDP*). Continuing from the earlier example, and using entry (3, 12) as the support entry, HDP yields the result shown in Figure 4.3b. Each entry of the upper left portion of the table represents the sum of the sphericity values at that entry resulting from the backward dynamic programming procedure. Likewise, each entry of the lower right portion of the table represents the sum of the sphericity values at that entry resulting from the forward dynamic programming procedure. The resulting maximum value path is shown in Figure 4.3c.

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, it indicates the similarity between a model and a scene landmark; 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 thus require that the entries along the path must be above a certain threshold to be considered as possible matches. A threshold of 0.7 is used as it provides reasonably good results. In the above example shown in Figure 4.3, 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 Figure 4.3 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 the adjacent landmarks to a scene landmark and the adjacent landmarks, a high sphericity value that is close to 1 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 0.95, its adjacent diagonal entries will also be considered as matches. In Figure 4.3, since all entries that are considered as matches between the model and the scene

landmarks have sphericity value greater than 0.95, 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.

The overall matching scheme between a sequence of model landmarks and a sequence of scene landmarks can be summarized by the following:

(1) Construct the table of compatibility between the sequence of model landmarks and the sequence of scene landmarks.

(2) Find the support entry.

(3) Perform HDP by switching between backward and forward dynamic programming. The backward procedure treats the support entry as a destination point and traces backward using the set of transition rules described earlier. The forward procedure treats the support entry as a starting point and advances forward using the set of transition rules also described earlier. At each stage, the procedure having a larger average sphericity proceeds one stage. The procedure stops when the combined path of both the backward and the forward procedure covers the range of either all the model landmarks or all the scene landmarks, i.e., either all the rows or all the columns of the table. The resulting path is known as the maximum value path.

(4) Find the entries along the maximum value path that result from diagonal transitions and are greater than 0.7. These entries are considered as possible matches between the model and the scene landmarks indicated by the indices of the entries.

(5) Isolated entries having no adjacent diagonal entries are nullified.

(6) Check for entries that have sphericity values greater than 0.95. The immediate adjacent diagonal entries of such high valued entries are then considered as matches.

4.3.3. Location Estimation and Matching Verification

After determining the landmarks of a model that match well with those of scene landmarks by HDP discussed in the previous section, we shall next estimate the location of the model object in the scene, and verify whether the hypothesis that this model object is in the scene is true. 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 matched scene landmarks in a least squares sense. A score based on the least squared error of the mapping is used to quantify the overall goodness of match between the model and the scene.

Let k be the number of pairs of the model and scene landmarks that match with each other,

$\{(x_1, y_1), (x_2, y_2), \cdots, (x_k, y_k)\}$ be the coordinates of the set of matched model landmarks, and

$\{(u_1, v_1), (u_2, v_2), \cdots, (u_k, v_k)\}$ be the coordinates of the set of the corresponding matched scene landmarks.

We want to find a coordinate transformation,

$$\begin{bmatrix} \hat{u} \\ \hat{v} \end{bmatrix} = \begin{bmatrix} a & b \\ -b & a \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} , \tag{4.1}$$

with the scale factor $= \sqrt{(a^2 + b^2)}$,

the angle of rotation $= \tan^{-1}(\frac{b}{a})$,

the translation coefficients $= \begin{bmatrix} e \\ f \end{bmatrix}$,

such that

$$\epsilon = \sum_{i=1}^{k} {e_{u_i}}^2 + {e_{v_i}}^2 \tag{4.2}$$

where $e_{u_i} = |\hat{u_i} - u_i| = |ax_i + by_i + e - u_i|$

$$e_{v_i} = |\hat{v_i} - v_i| = |-bx_i + ay_i + f - v_i|$$

is minimized. Note that ϵ is the least squared error of the transformation. By finding the partial derivatives of Equation 4.2 with respect to each coefficient of the coordinate transformation described by Equation 4.1, we can obtain the following coefficients of the least squares coordinate transformation:

$$\begin{bmatrix} a \\ b \\ e \\ f \end{bmatrix} = T^{-1} \begin{bmatrix} \sum_{i=1}^{k}(u_i x_i + v_i y_i) \\ \sum_{i=1}^{k}(u_i y_i - v_i x_i) \\ \sum_{i=1}^{k} u_i \\ \sum_{i=1}^{k} v_i \end{bmatrix}$$

where $$T = \begin{bmatrix} \sum_{i=1}^{k}(x_i^2+y_i^2) & 0 & \sum_{i=1}^{k} x_i & \sum_{i=1}^{k} y_i \\ 0 & \sum_{i=1}^{k}(x_i^2+y_i^2) & \sum_{i=1}^{k} y_i & -\sum_{i=1}^{k} x_i \\ \sum_{i=1}^{k} x_i & \sum_{i=1}^{k} y_i & k & 0 \\ \sum_{i=1}^{k} y_i & -\sum_{i=1}^{k} x_i & 0 & k \end{bmatrix}.$$

Continuing from the earlier example, the wire stripper is mapped into the scene, as shown in Figure 4.4, by the least squared coordinate transformation derived from the matching 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 squared coordinate transformation can be used as an additional parameter for verifying the match.

The above least squared error 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. To account for the overall goodness of match between the model and the scene, we use the following heuristic measure which penalizes incomplete matching of the landmarks of the model:

$$\epsilon' = \begin{cases} (1.0+(\frac{n-2}{k-2})\log_2(\frac{n-2}{k-2}))\bar{\epsilon} & \text{for } k \geq 3, \\ \infty & \text{for } k = 0,1,2. \end{cases} \qquad (4.3)$$

where n is the total number of landmarks of the model,

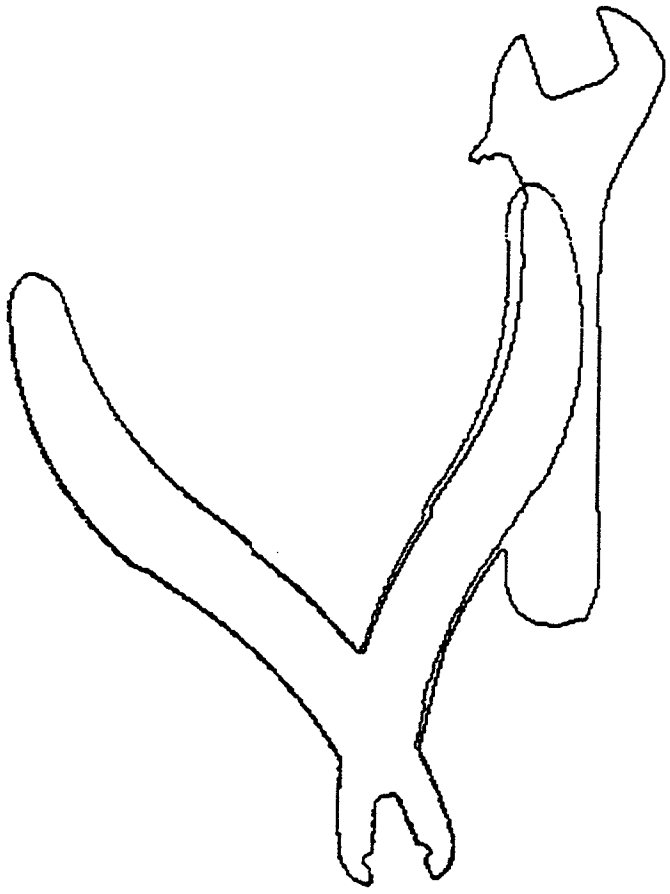


Figure 4.4. The result of mapping the wire stripper into the scene by the least squared coordinate transformation.

k is the number of the model landmarks that match with the scene landmarks, and

$\bar{\epsilon} = \frac{\epsilon}{k(\text{scale factor})}$, i.e., $\bar{\epsilon}$ is the normalized least squared error.

The heuristic measure, ϵ', which can be regarded as the error measure for the overall goodness of match between the model and scene, is referred to as the *match error*. If only one or two model landmarks match with 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 into another set. We consider such cases where only 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. Thus, in Equation 4.3, we have $\epsilon' = \infty$ when k=0,1,2, and we use $\frac{n-2}{k-2}$ instead of $\frac{n}{k}$. Note that when $k=n$, $\epsilon' = \bar{\epsilon}$; i.e., no penalty is added to the normalized least squared error when all model landmarks match with those in the scene. The penalty is higher if k is smaller. According to [Gal68], if $\frac{k-2}{n-2}$ is considered as the probability of the event that k of the n model landmarks match with those in the scene, $\log_2(\frac{n-2}{k-2})$ can be interpreted as the uncertainty or the *self-information* of the event. The term, $(\frac{n-2}{k-2})$, in front of the self-information can be thought of as the penalty incurred per amount of uncertainty.

In the earlier example, since all model landmarks match with those in the scene, the match error value of 0.62 is the same as the normalized least squared error. The hypothesis of the model in the scene is finally determined by the value of the match error — a small error verify the hypothesis while a large error 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 is considered incorrect. In our study, this threshold is set empirically.

4.3.4. Computational Complexity of Hopping Dynamic Programming

In the landmark matching task, we first determine the support entry of the table of compatibility. From earlier discussion, the support entry is the one where the sum of the entry and its two immediate adjacent diagonal entries is maximum. If the table has n model landmarks and m scene landmarks, it will require $3nm$ additions and nm comparisons to determine the support entry.

To determine the computational complexity of HDP, we shall determine the complexity of both the forward and the backward procedures. Consider a 4×4 table with row indices 1, 2, 3, and 4, and column indices 1, 2, 3, and 4. It requires 9 additions and 24 comparisons to find the accumulated sum of sphericity values for each entry of the table using the forward procedure of HDP. These numbers for additions and comparisons are determined by the transition rules of the forward procedure (see Section 4.3.2). Note that each move according to Rule (1) requires 1 addition and 2 comparisons, Rule (2) 1 comparison, Rule (3) 1 comparison, and Rule (4) which is the initial condition requires no computation. In this example of the table, entry (1, 1) is the initial condition. Entries along the first row of the table are determined by transition Rule (3), and entries along the first column by transition Rule (2). Besides the initial entry, there are 3 entries along the first row as well as along the first column of the table, and thus it requires 6 comparisons to determine the accumulated sum of sphericity values for these entries. The remaining 9 (3×3) entries which are determined by transition Rule (1) require 9 additions and 18 comparisons. Thus, a total of 9 additions and 24 comparisons are required. Generalizing this analysis, we can conclude that if the table is $n \times n$, it will require $2(n-1)$ comparisons to determine the values for entries along the first row and the first column, and $(n-1)^2$ additions and $2(n-1)^2$ comparisons for the remaining entries; a total of $(n-1)^2$ additions and $2n(n-1)$ comparisons. By the same reasoning, the backward procedure has the same computational complexity as the forward procedure.

Let $N=\min(n, m)$ denote the minimum between the number of model landmarks and the number of scene landmarks. Since HDP stops when the range of either all the model landmarks or all the scene landmarks is covered, only an $N \times N$ portion of the $n \times m$ table of compatibility will be used in the procedure. Of the $N \times N$ portion of the table, the upper left portion, which we denote as having a size of $p \times p$, results from the backward procedure, and the lower right portion of size $(N-p+1) \times (N-p+1)$ from the forward procedure. The computational complexity is proportional to the total number of entries that are covered by HDP. The total number of entries is $p^2+(N-p+1)^2$.

By finding the derivative of the total number of entries with respect to p, the value of $p=\frac{N+1}{2}$ yields the least number of entries. In this case, when N is odd, by adding the number of computations required for both the forward and backward procedures of HDP, a total of

$$\frac{1}{2}(N-1)^2 \text{ additions and } (N^2-1) \text{ comparisons} \tag{4.4}$$

are required. If N is even, the complexity is least expensive when one procedure covers $\left(\left\lceil\frac{N+1}{2}\right\rceil\right)^2$ and the other $\left(\left\lfloor\frac{N+1}{2}\right\rfloor\right)^2$, where $\lceil . \rceil$ is the ceiling operator, i.e., the smallest integer greater than or equal to the argument, and $\lfloor . \rfloor$ is the floor operator, i.e., the largest integer less than or equal to the argument. In this case, HDP requires

$$(\frac{N^2}{2}-N+1) \text{ additions and } N^2 \text{ comparisons .} \tag{4.5}$$

The computation is the most expensive when the procedure covers all $N \times N$ entries, in which case it requires

$$(N-1)^2 \text{ additions and } 2N(N-1) \text{ comparisons .} \tag{4.6}$$

This happens when either only a forward or only a backward procedure is used.

The splitting of the $N \times N$ table into a portion governed by the forward procedure and the other by the backward procedure is problem dependent. The computational complexity of HDP is thus bounded between the amount of computation defined by Equation 4.6 and that of Equation 4.4 or 4.5. Additional computational overhead is required by HDP to decide to which procedure to switch. This overhead requires N comparisons and N divisions. Each division is used to calculate the average sphericity value mentioned in Section 4.3.2. In general, as will be seen in examples presented in the next section and the next chapter, each object is usually represented by no more than 100 landmarks. It is thus computationally inexpensive to determine matches between landmarks of a model and a scene.

4.4. Experimental Results

We shall present three examples of a scene which contain overlapping tools. Further experimental results which takes into account the effect of noise and larger occlusion will be discussed in the next chapter.

Consider again the scene shown in Figure 4.2, the results of performing the landmark matching task between the scene and each of the models shown in Figures 3.34-3.38 are summarized in Table 4.1. Models that match well with the objects in the scene are those with the smallest match errors. Though the

Table 4.1.
The summary of the results of matching a
library of objects with the scene shown in Figure 4.2.

Models	Model figure numbers	Total Number of Model land-marks	Number of matched model land-marks	Match Error
wrench	3.35	6	6	1.98
needle-nose plier	3.36	4	2	∞
wire cutter	3.34	6	5	7.39
specialty plier	3.37	6	2	∞
wire stripper	3.38	6	6	0.62

wire cutter is not in the scene, the match error between the wire cutter and the scene is quite small. This is because the scene contains the wire stripper, and the relative positions of the landmarks of the wire stripper are similar to those of the wire cutter. Figures 4.5-4.7 show the results of mapping other models into the scene.

Figure 4.8 shows another example of a scene in which the landmarks are extracted based on the cardinal curvature points using ω=20. Using the same library of models as the previous example, the results of matching each model with the scene are summarized in Table 4.2. Again, models that match well with the objects in the scene are those with the smallest match errors. The results of mapping the two correctly matched models into the scene are shown in Figures 4.9-4.10.

The last example of a scene is shown in Figure 4.11, where the landmarks are obtained by the curvature guided polygonal approximation using ω=10, and a collinear factor of 15. A library of models in which landmarks are obtained by the same method and using the same parameters are shown in Figures 3.13-3.17. Each of the models and the scene contains more landmarks than the previous examples. The results of matching are summarized in Table 4.3. Note again that correct matches between the models and the scene have the smallest match errors. Figures 4.12-4.15 show the results of mapping some

of the models into the scene. Note that some of the objects in the scene contains extraneous and missing landmarks due to occlusion.

4.5. Summary

We have reviewed recent work on partial shape recognition, and compared it with our approach. Our approach is unique and efficient — we use landmarks as the shape features, sphericity as a shape measure, and hopping dynamic programming for matching the landmarks. Instead of computing several feature values to quantify the similarity between two features, we use a single shape measure, sphericity, which is easy to compute. The landmark matching task is computationally less expensive than other feature matching tasks which involve iterative procedures. We have presented some experimental results. Further experimental results will be presented in the next chapter.

Table 4.2.
The summary of the results of matching a
library of objects with the scene shown in Figure 4.8.

Models	Model figure numbers	Total Number of Model land-marks	Number of matched model land-marks	Match Error
wrench	3.35	6	5	2.89
needle-nose plier	3.36	4	3	14.97
wire cutter	3.34	6	3	8.01
specialty plier	3.37	6	6	2.16
wire stripper	3.38	6	3	11.65

Table 4.3.
The summary of the results of matching a
library of objects with the scene shown in Figure 4.11.

Models	Model figure numbers	Total Number of Model landmarks	Number of matched model landmarks	Match Error
wrench	3.13	10	7	1.11
needle-nose plier	3.14	10	8	3.72
wire cutter	3.15	12	11	1.28
specialty plier	3.16	8	3	10.23
wire stripper	3.17	10	2	∞

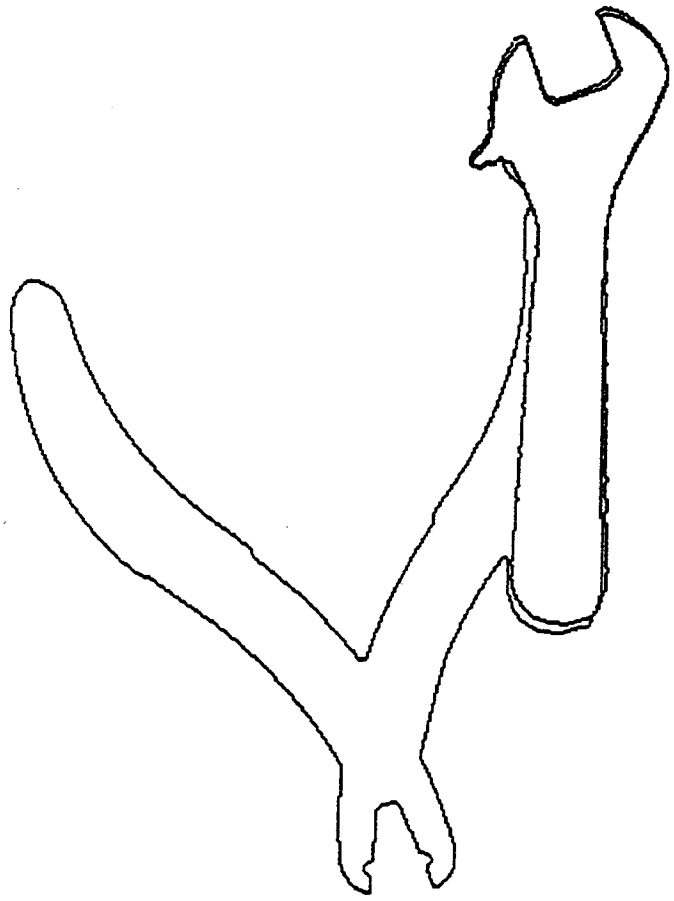


Figure 4.5. The result of mapping the wrench into the scene shown in Figure 4.2.

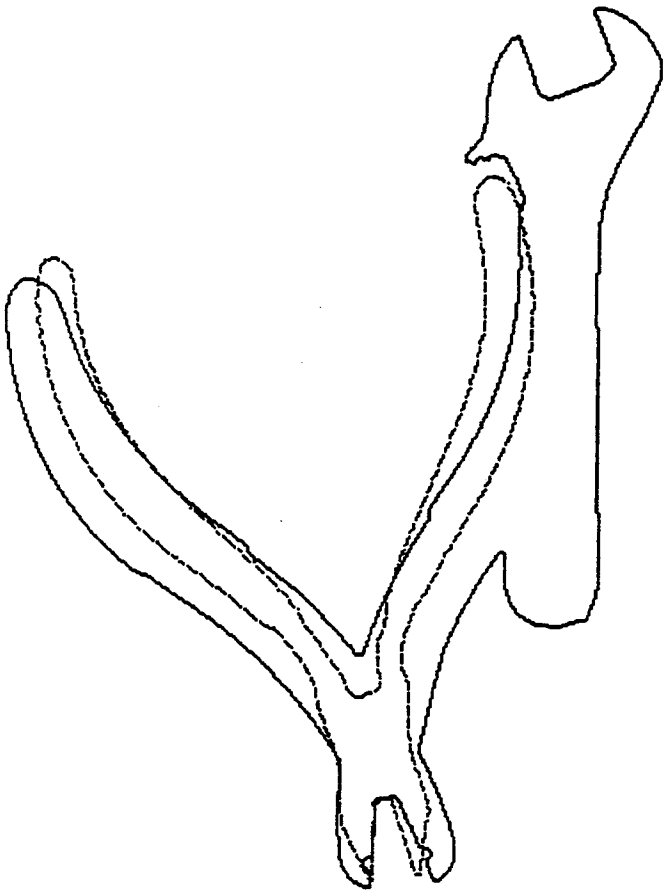


Figure 4.6. The result of mapping the wire cutter into the scene shown in Figure 4.2.

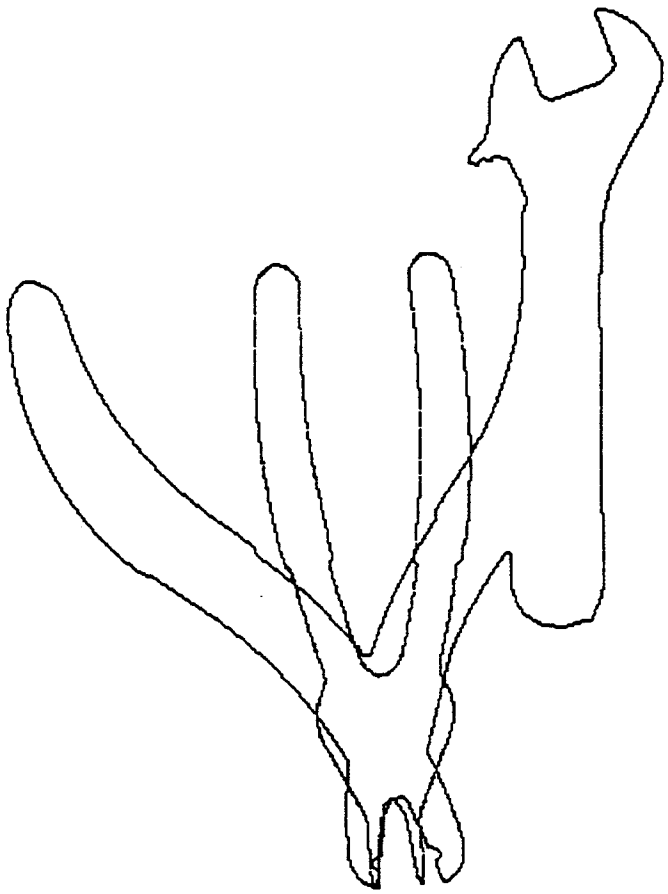


Figure 4.7. The result of mapping the specialty plier into the scene shown in Figure 4.2.

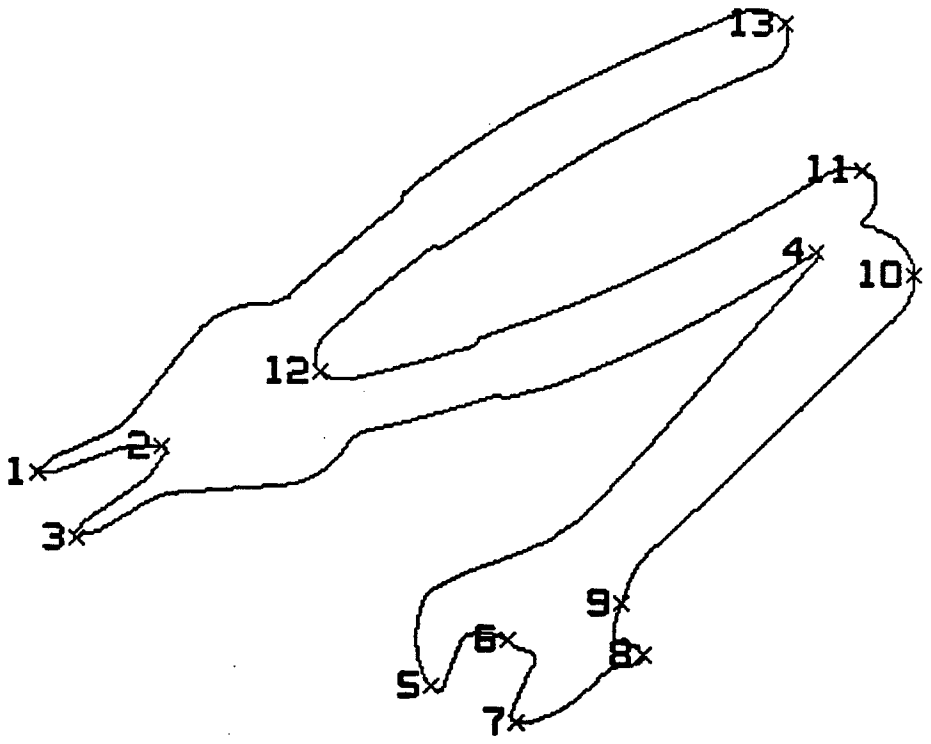


Figure 4.8. A scene which consists of a .specialty plier and a wrench overlapping each other. Each scene landmark is labeled and indicated by an "X."

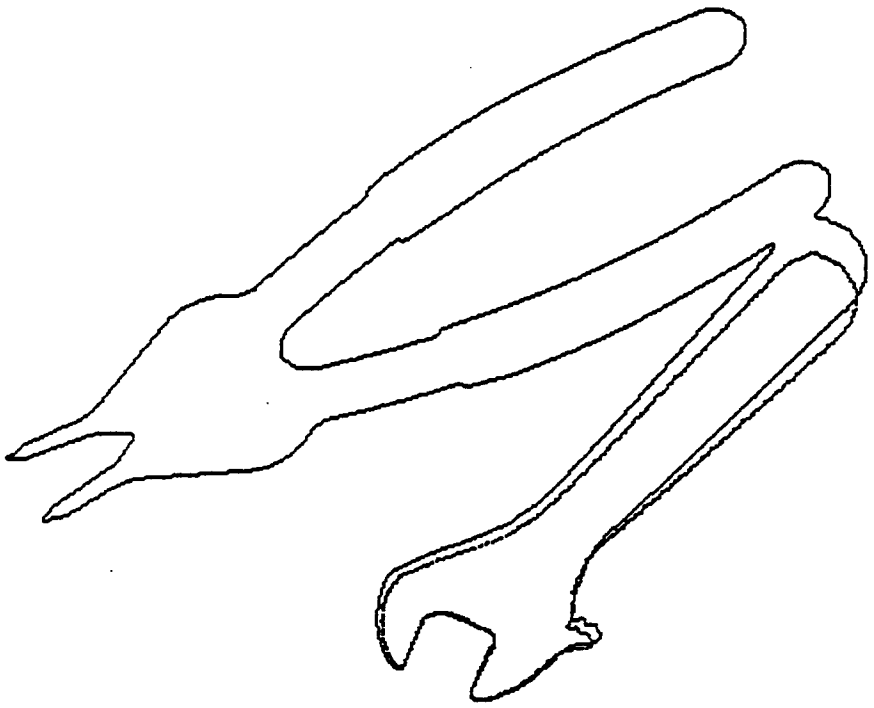


Figure 4.9. The result of mapping the wrench into the scene shown in Figure 4.8.

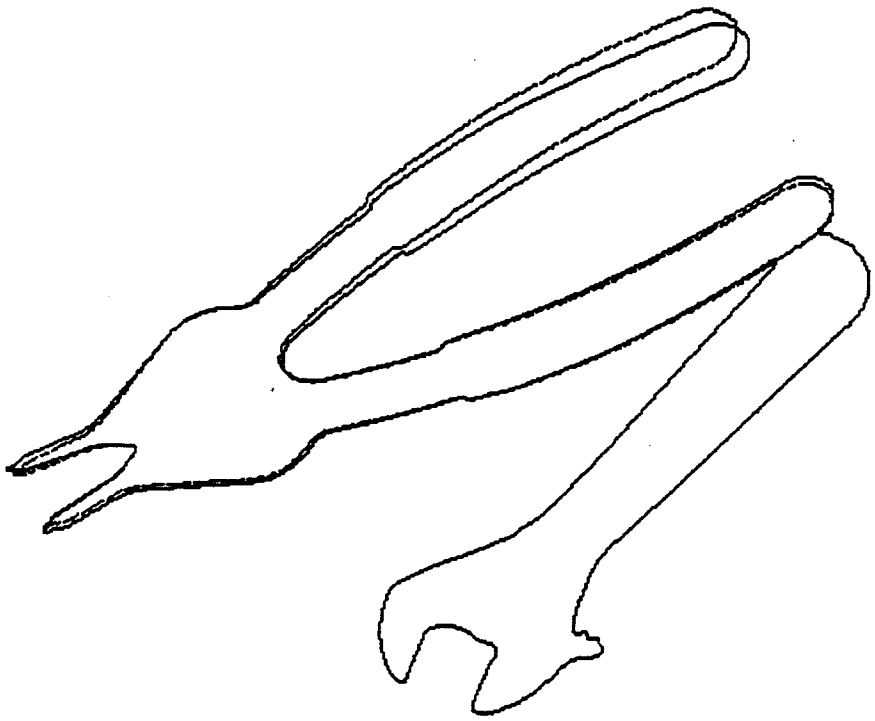


Figure 4.10. The result of mapping the specialty plier into the scene shown in Figure 4.8.

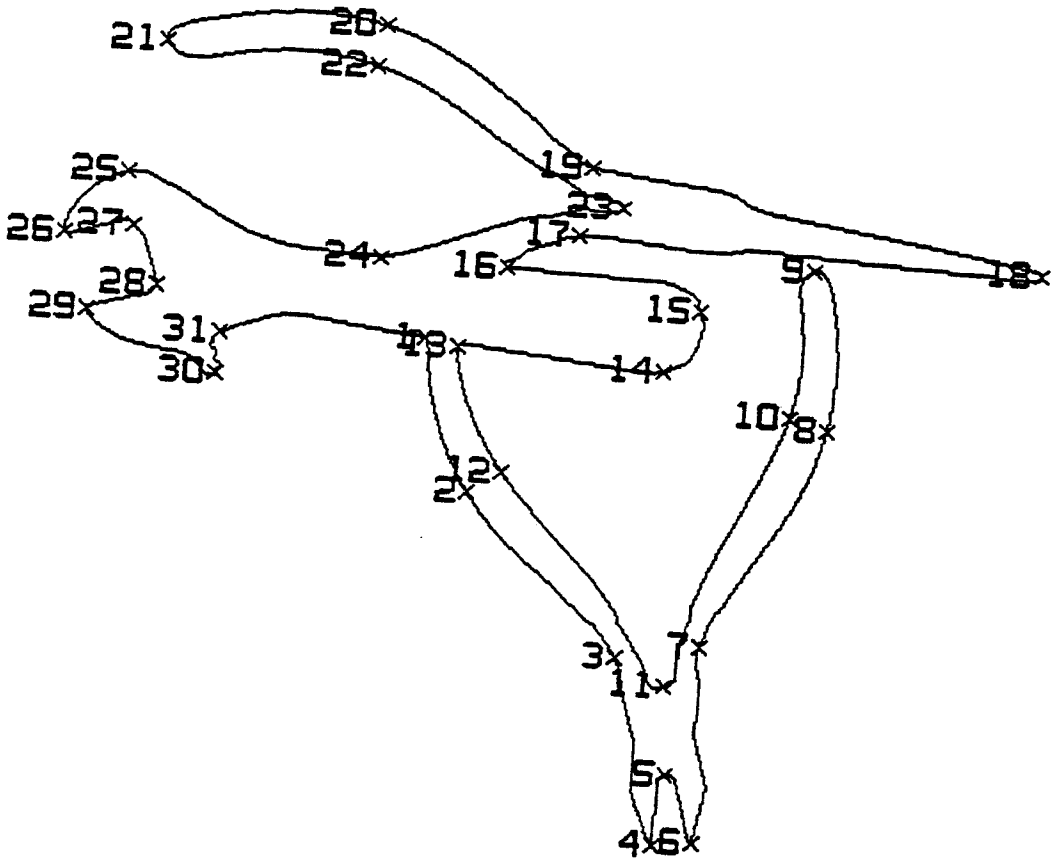


Figure 4.11. A scene which consists of a wire cutter, a wrench, and a needle-nose plier overlapping each other. Each scene landmark is labeled and indicated by an "X."

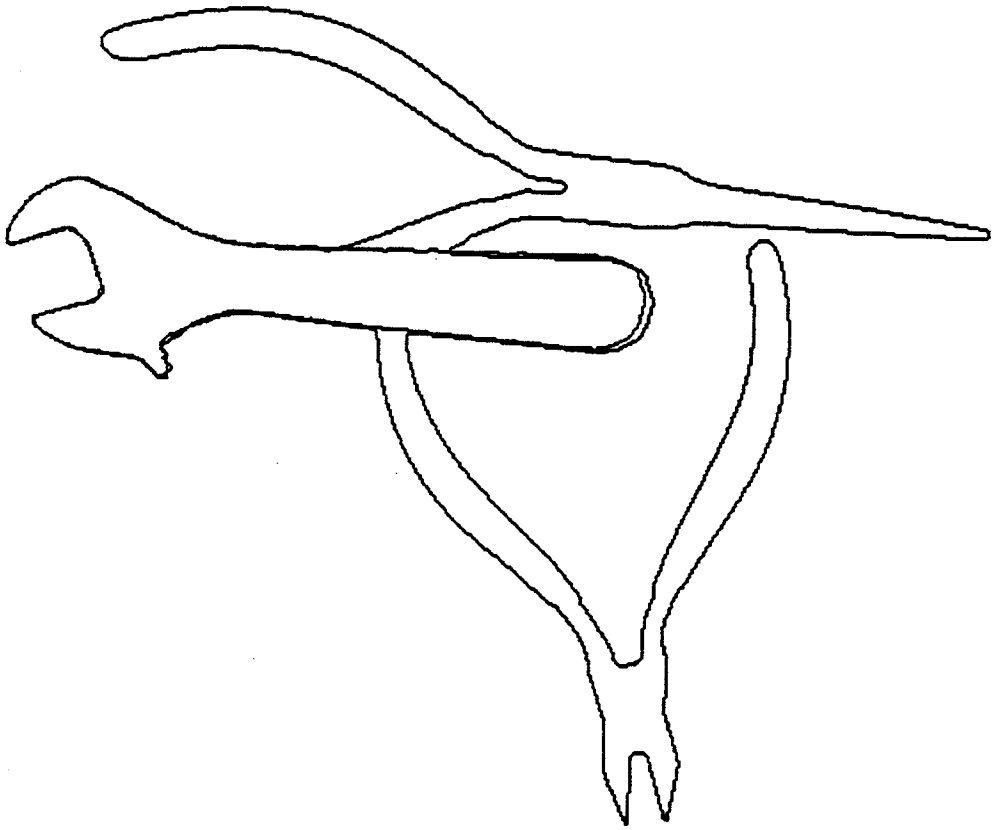


Figure 4.12. The result of mapping the wrench into the scene shown in Figure 4.11.

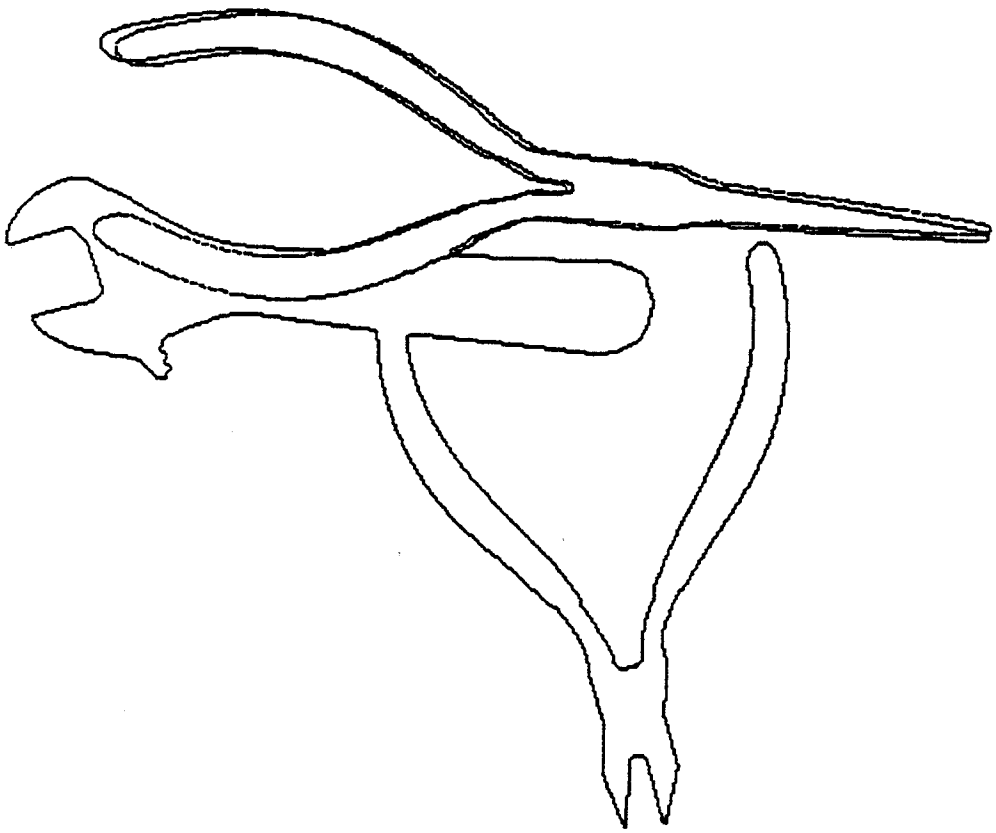


Figure 4.13. The result of mapping the needle-nose plier into the scene shown in Figure 4.11.

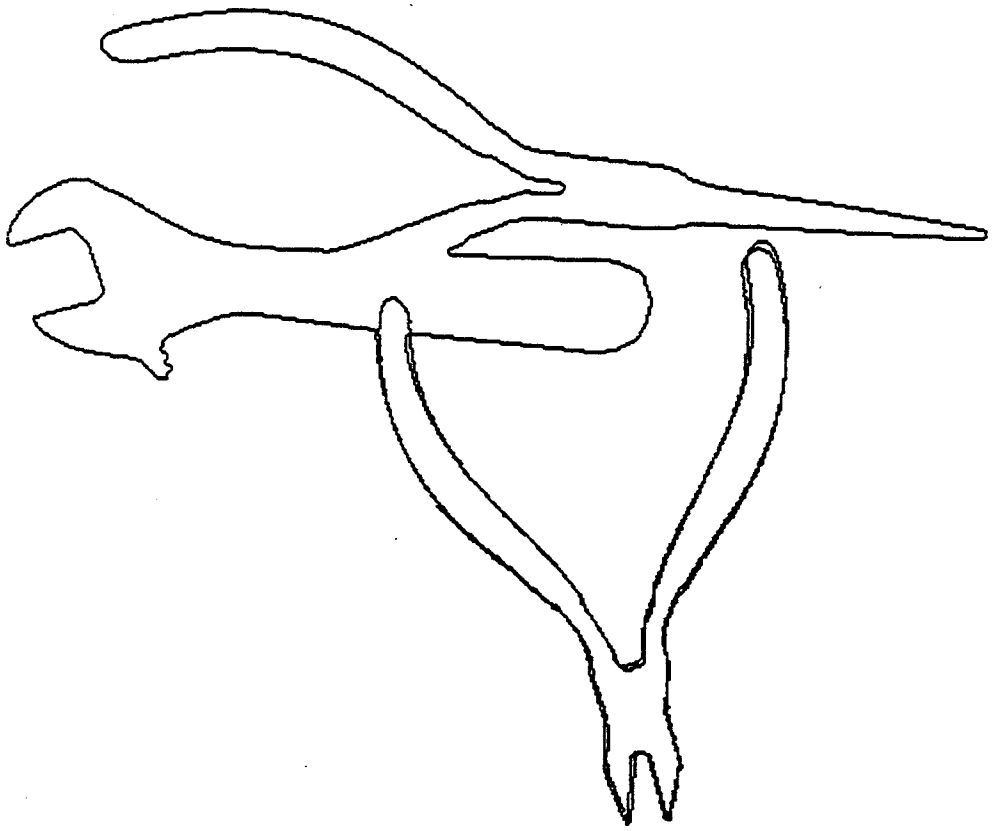


Figure 4.14. The result of mapping the wire cutter into the scene shown in Figure 4.11.

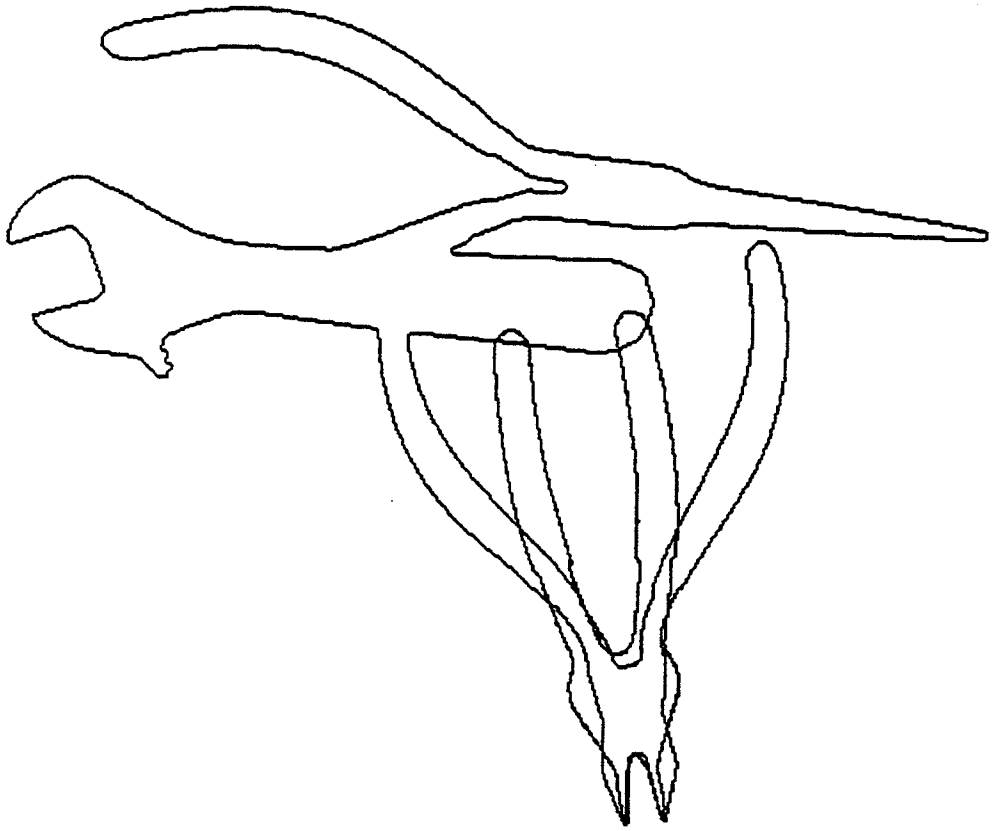


Figure 4.15. The result of mapping the specialty plier into the scene shown in Figure 4.11.

CHAPTER 5

FURTHER EXPERIMENTAL RESULTS

5.1. Introduction

In Chapters 1 through 4, we have presented the overall scheme along with some experimental results of landmark-based shape recognition. In this chapter, we shall present further experimental results to demonstrate the effectiveness of the approach with respect to noise and larger occlusion. Four examples will be studied. An example on noise effects will be presented in Section 5.2, two examples on larger occlusion will be presented in Section 5.3, and one example on the combined effects of noise and occlusion will be presented in Section 5.4. We shall summarize the chapter in Section 5.5.

5.2. Experiments with Noisy Data

We consider an image of a spacecraft shown in Figure 5.1a. The gray level value of the object region is 160, and the background is 96. The silhouette of the spacecraft is shown in Figure 5.1b, the contour in Figure 5.1c, and the landmarks in Figure 5.1d. The landmarks are extracted based on the cardinal curvature points using $\omega = 20$. We shall consider the image shown in Figure 5.1a as the image of a model object. To simulate the effects of noisy data, a zero mean i.i.d. Gaussian random variable is added to each pixel of the noiseless image. The noisy 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. The landmarks are also extracted based on the cardinal curvature points. We shall consider the noisy image as the image of a 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:

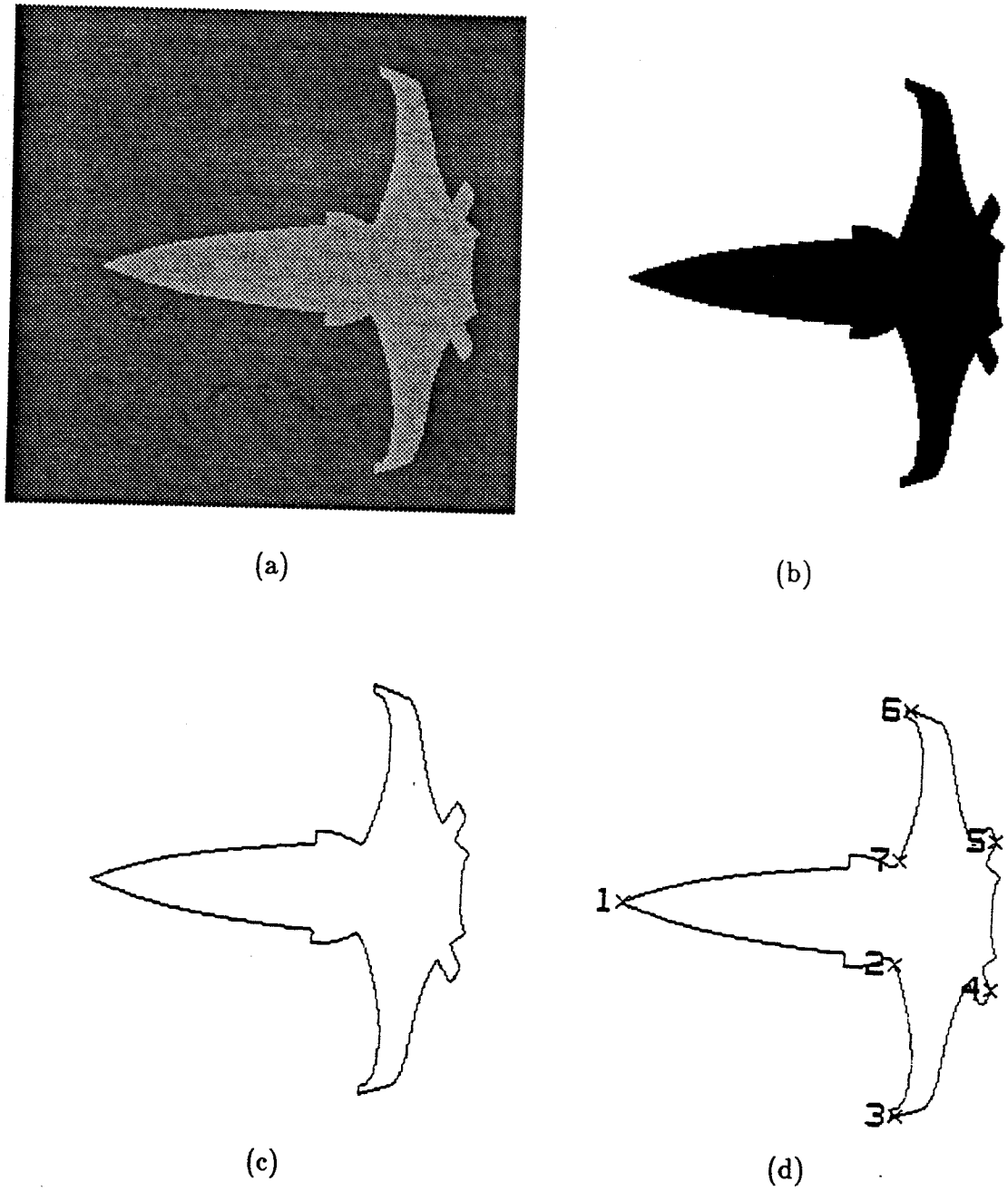


Figure 5.1. Experiments with noisy data: noiseless case. (a) A 256×256 gray level image of a spacecraft. (b) The silhouette of the spacecraft. (c) The contour of the spacecraft. (d) Landmarks of the spacecraft based on the cardinal curvature points. Each landmark is labeled and indicated by an "×."

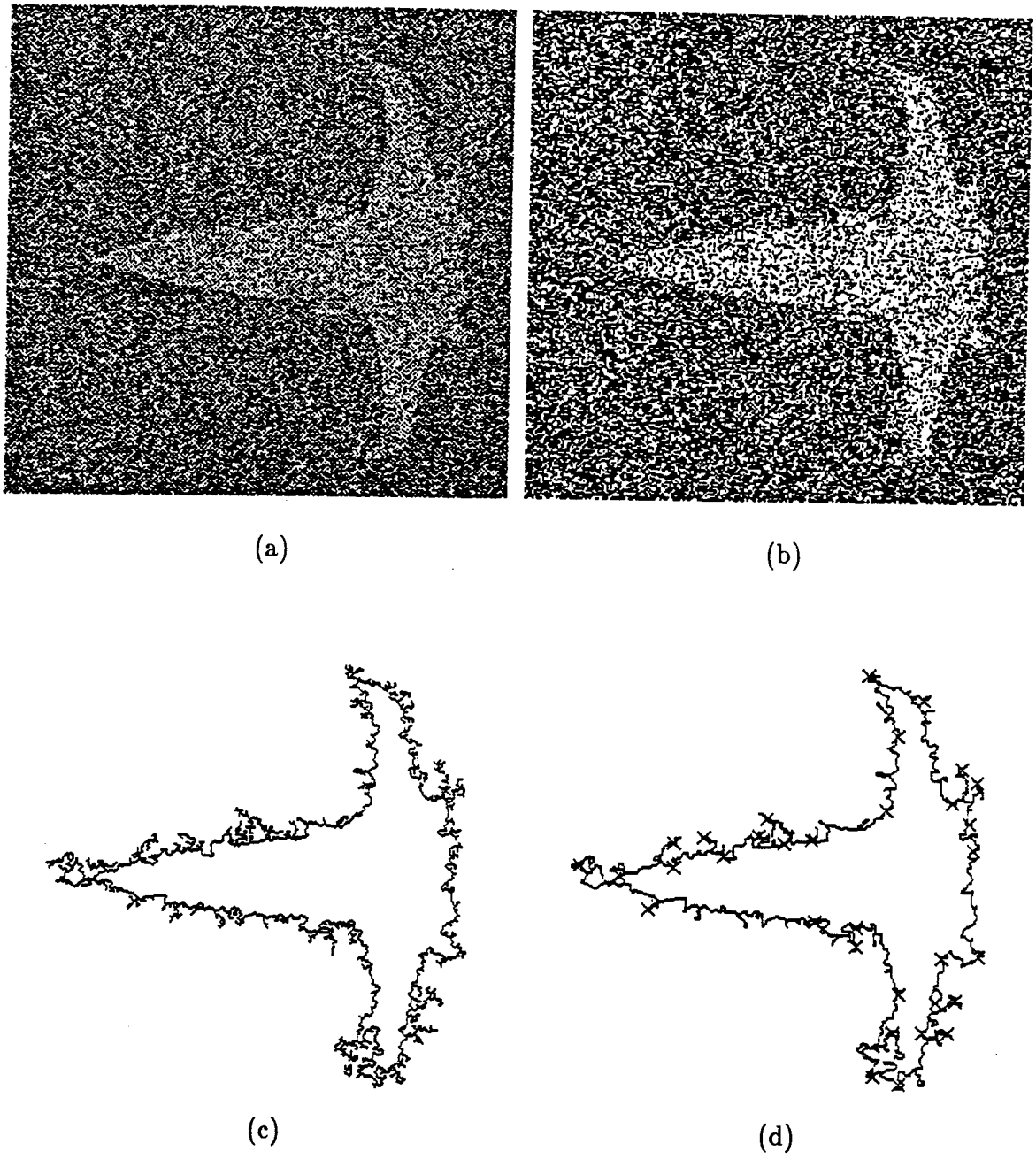


Figure 5.2. Experiments with noisy data: SNR=0dB. (a) The noisy image of the spacecraft having a SNR=0dB . (b) The result of thresholding the noisy image. (c) The corresponding object contour in the noisy image. (d) Extracted landmarks. Each landmark is labeled and indicated by an "×."

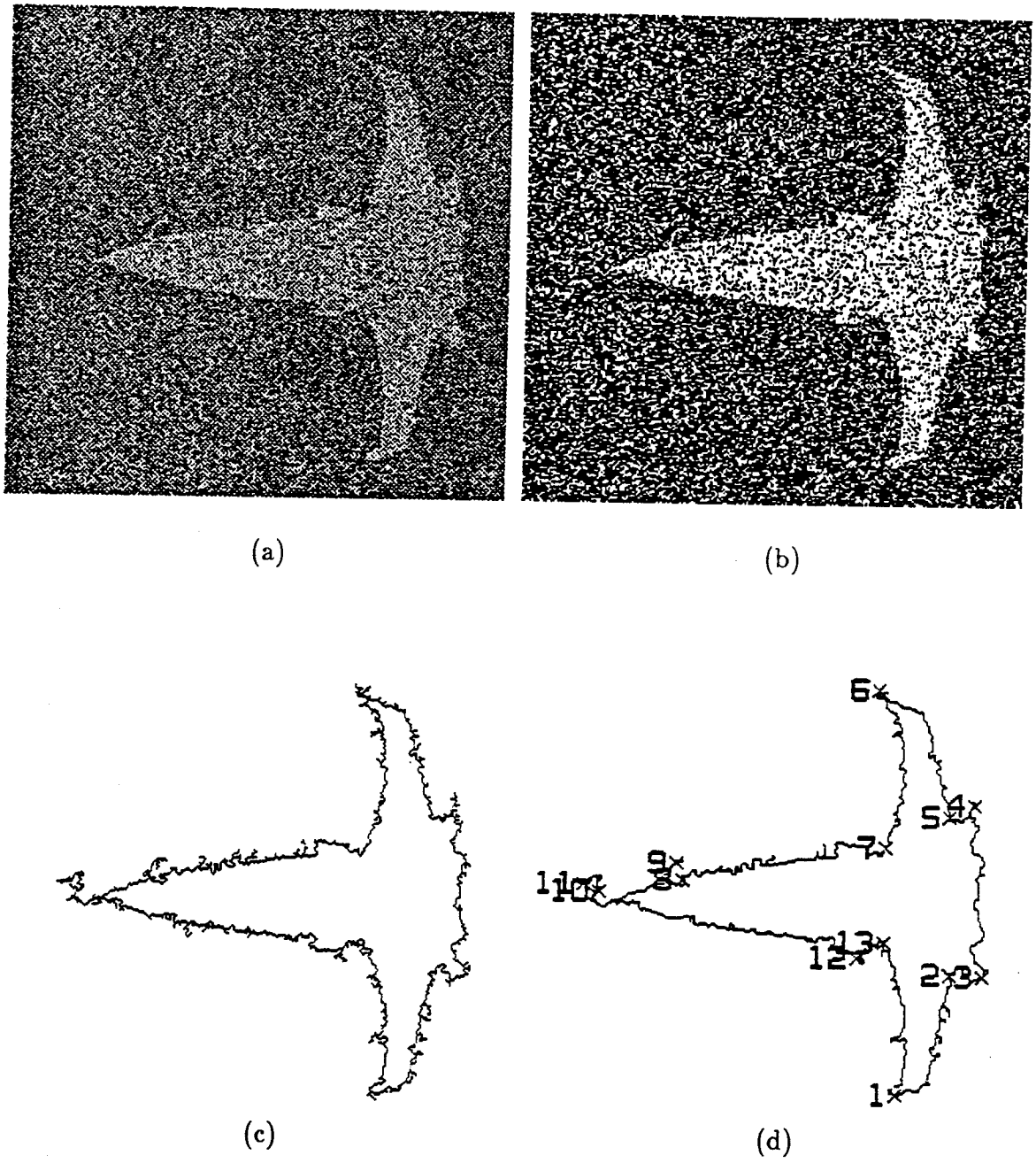


Figure 5.3. Experiments with noisy data: SNR=3dB. (a) The noisy image of the spacecraft having a SNR=3dB . (b) The result of thresholding the noisy image. (c) The corresponding object contour in the noisy image. (d) Extracted landmarks. Each landmark is labeled and indicated by an "×."

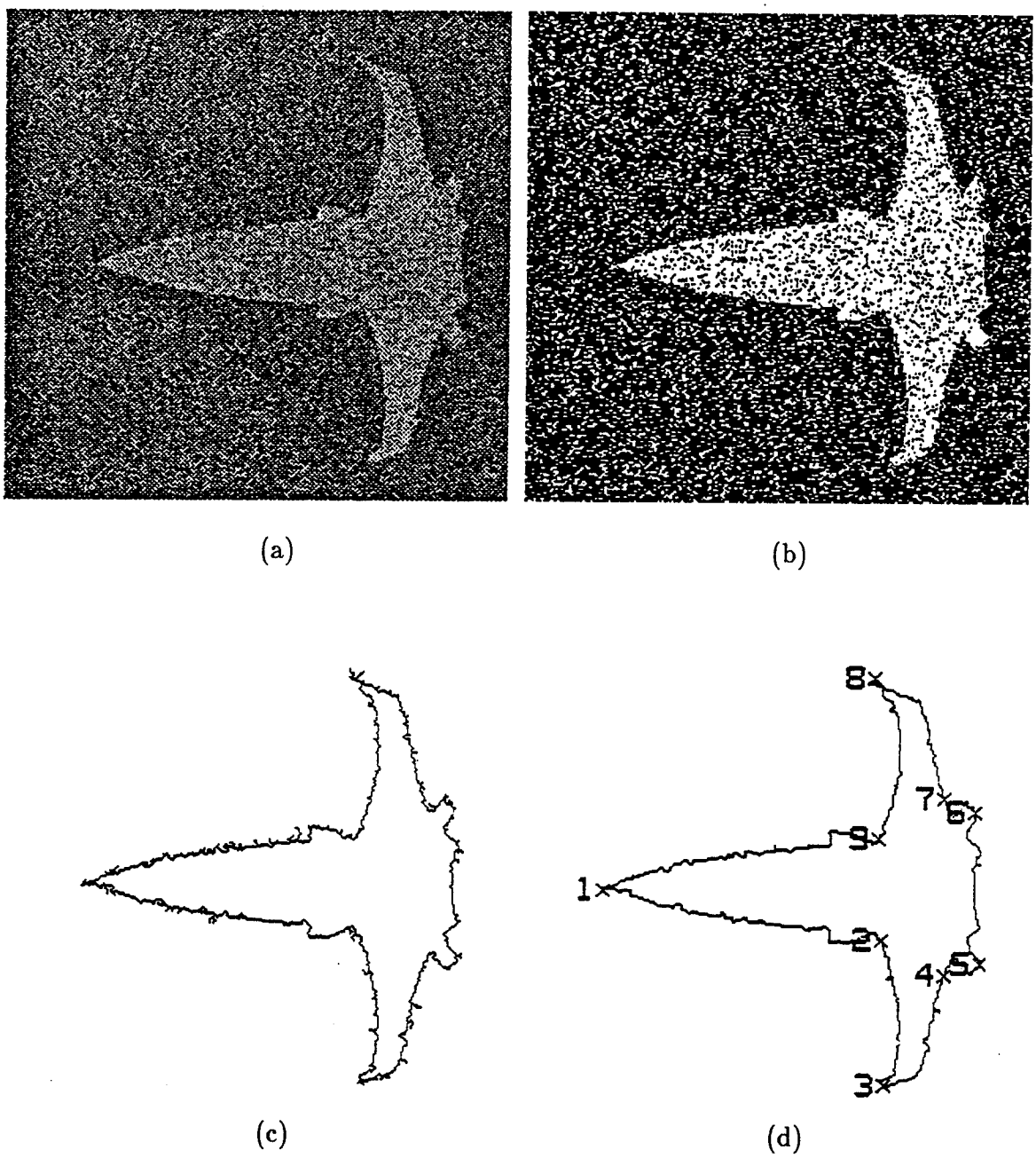


Figure 5.4. Experiments with noisy data: SNR=6dB. (a) The noisy image of the spacecraft having a SNR=6dB . (b) The result of thresholding the noisy image. (c) The corresponding object contour in the noisy image. (d) Extracted landmarks. Each landmark is labeled and indicated by an "×."

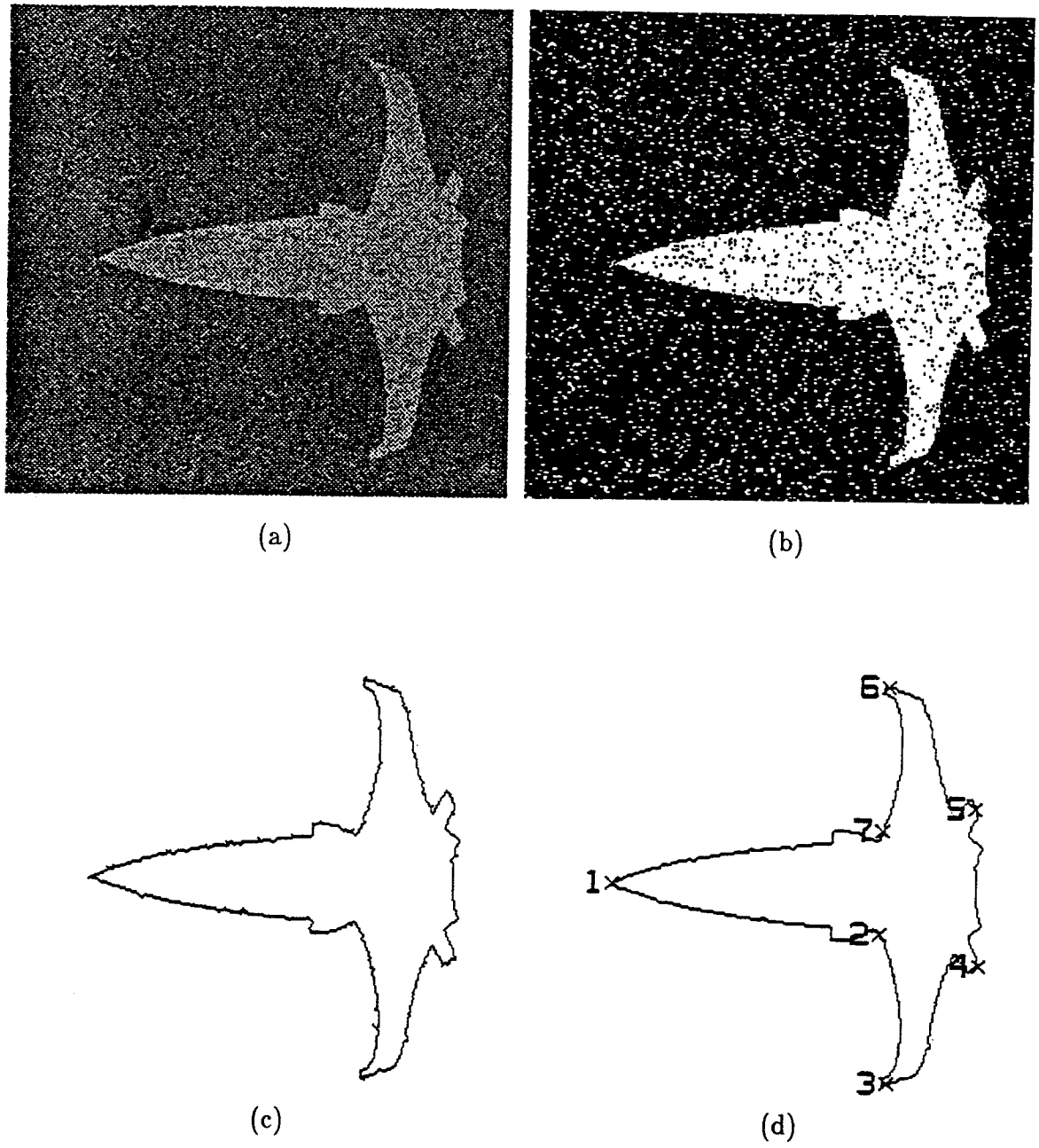


Figure 5.5. Experiments with noisy data: SNR=10dB. (a) The noisy image of the spacecraft having a SNR=10dB . (b) The result of thresholding the noisy image. (c) The corresponding object contour in the noisy image. (d) Extracted landmarks. Each landmark is labeled and indicated by an "×."

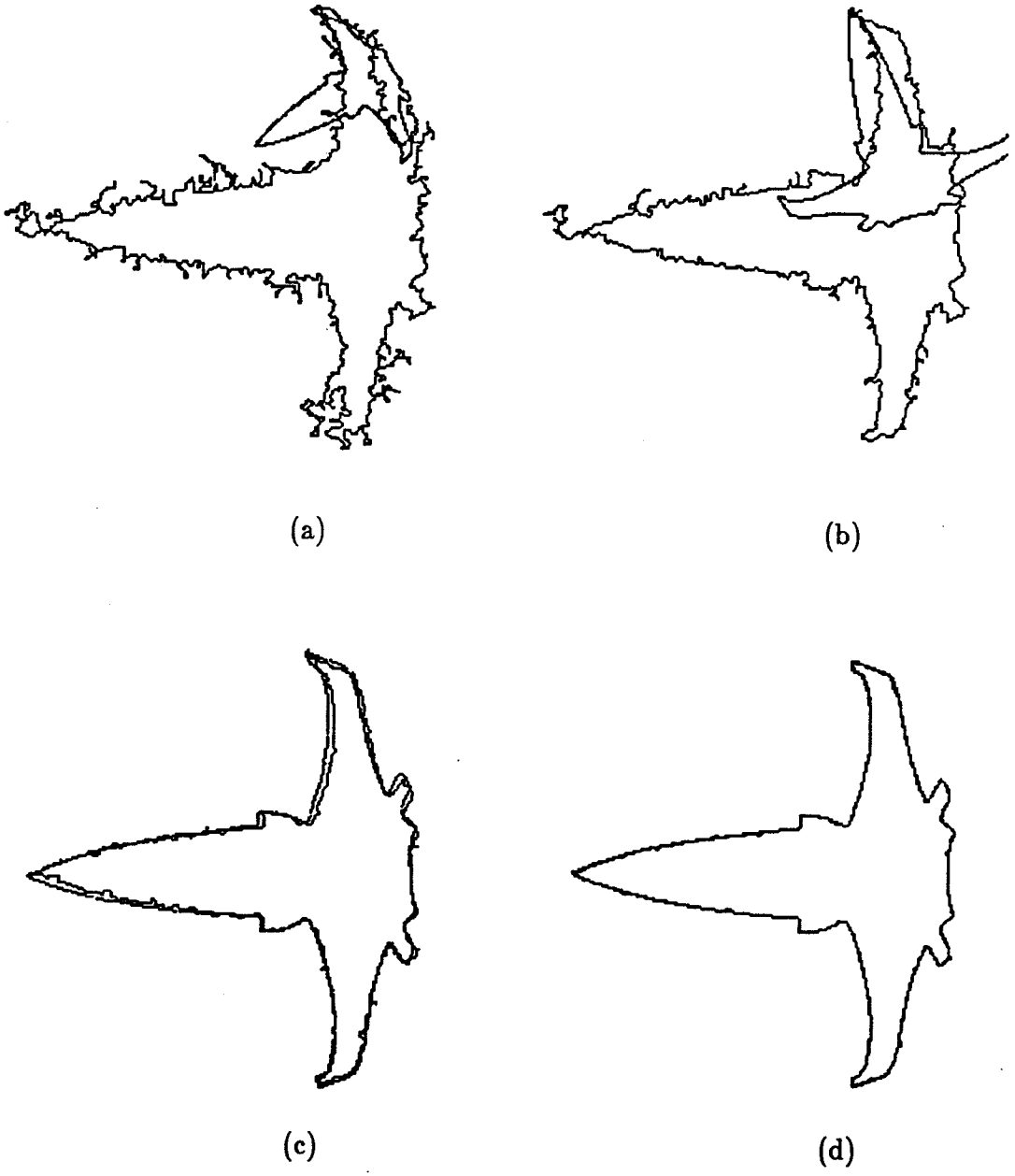


Figure 5.6. The results of mapping the model into the scenes with SNR=0, 3, 6, 10 dB are shown in (a), (b), (c), and (d), respectively.

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

where 64 is the difference between the gray level values of the object region and the background.

Images having SNR's of 0 dB, 3 dB, 6 dB, and 10 dB are shown in Figures 5.2a, 5.3a, 5.4a, and 5.5a, respectively. Their corresponding thresholded images are shown in Figures 5.2b, 5.3b, 5.4b, and 5.5b, respectively; their corresponding object contours are shown in Figures 5.2c, 5.3c, 5.4c, and 5.5c, respectively; their corresponding extracted landmarks are shown in Figures 5.2d, 5.3d, 5.4d, and 5.5d, respectively. Notice that when the SNR is low, the landmark extractor produces many erroneous and extraneous landmarks. As compared to the model which has only 7 landmarks, 33 landmarks are extracted from the scene which has a SNR = 0 dB. The match error between the model and the 0 dB scene is 36.53. The result of mapping the model contour into the scene is shown in Figure 5.6a. The match errors between the model and the scenes with SNR = 3 dB, 6 dB, and 10 dB are 45.13, 2.55, and 0.66, respectively. The results of mapping the model contour into these scenes are shown in Figure 5.6b, 5.6c, and 5.6d, respectively.

As seen from these experiments, when too many erroneous landmarks occur and the sequential order of the landmarks is lost, the local structures of the scene landmarks become completely different from those of the model landmarks. In this case, the landmark matching task fails to determine the correct matches. As compared with the model landmarks, when a part of the sequential order of the landmarks in the scene is still preserved, although there may have minor distortion in the landmark locations, the landmark matching task can successfully detect correct matches. In Figures 5.2d and 5.3d, the sequential order of the scene landmarks is totally lost as compared with that of the model, and hence the matching task fails. In Figure 5.4d, the locations of the landmarks along the object contour in the scene deviate a small amount from those of the model landmarks. In addition, part of the sequential order of the landmarks of the object in the scene has also been rearranged by two extraneous landmarks. However, the matching task can still correctly match model landmarks 6, 7, 1, 2, and 3 with scene landmarks 8, 9, 1, 2, and 3, respectively, yielding a small match error. It is seen from these experiments that the sequential order of the landmarks is important to the matching task, but minor distortion in landmark locations does not significantly degrade the performance.

5.3. Experiments with Occlusion

In Chapter 4, we have considered scenes that comprise of at most three overlapping objects. Before we consider more complicated scenes, we shall expand the library of model objects. In addition to the set of tools discussed in Chapter 3, we shall include the spacecraft, and the outlines of some islands shown in Figures 5.7-5.12. The islands, which are not man-made, have interesting and complicated shapes. Figures 5.7-5.12 show the silhouettes and the corresponding landmarks of the islands of Borneo, Halmahera, Luzon, Mindanao, New Guinea, and Sulawesi, respectively. The landmarks of the islands are extracted based on the cardinal curvature points, and the corresponding range of ω values that can be used for extracting landmarks of each island are summarized in Table 5.1. Notice that the outlines of the islands are very curvy, and the corresponding range of ω values that can be used for extracting their landmarks are smaller than those of the tools.

Table 5.1.
Range of ω values used to obtain the landmarks of the additional objects of the enlarged library based on cardinal curvature points

Models	Figures	Range of ω
spacecraft	5.1d	19.5-25.5
Borneo	5.7b	19.5-24
Halmahera	5.8b	20-23
Luzon	5.9b	14-19.5
Mindanao	5.10b	17.5-21.5
New Guinea	5.11b	18-21
Sulawesi	5.12b	19.5-29

Figure 5.13 shows an example of a scene which consists of four objects overlapping each other. Compared to their respective models, the wire stripper has been scaled by an area factor of 0.6, and the island of New Guinea by an area factor of 1.4. The spacecraft has been rotated by 45°. Landmarks in the scene are extracted based on the cardinal curvature points using ω=20. Compared to their respective model landmarks, one out of six of the landmarks

of the wire stripper, one out of seven of the landmarks of the spacecraft, one out of nine of the landmarks of Sulawesi, and five out of eleven of the landmarks of New Guinea are missing. With respect to each model, those landmarks in the scene not belonging 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 5.2. Notice that the models that correctly match the scene have the smallest match errors. Although the scene does not contain the wire cutter, the match error between the wire cutter and the scene is quite small. This is due to the fact that the scene contains the wire stripper and the relative locations of the landmarks of the wire cutter are quite similar to those of the wire stripper. The landmarks of the wire cutter are thus matched with those of the wire stripper in the scene. Figures 5.14-5.17 show the results of mapping the models that correctly match with the scene back into the scene.

Figure 5.18 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. The landmarks in the scene are also extracted based on cardinal curvature points using ω=20. 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. Again, with respect to each model, those landmarks in the scene not belonging 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 5.3. Also note that the models that correctly match the scene have the smallest match errors. Figures 5.19-5.24 show the results of mapping the models that correctly match with the scene back into the scene.

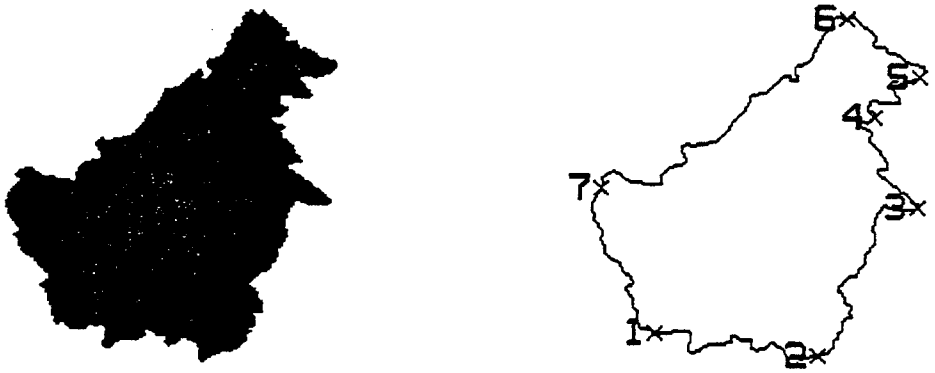
5.4. Experiments on The Combined Effects of Noise and Occlusion

In this section, we shall investigate the effectiveness of our landmark matching task with regard to the combined effects of noise and occlusion. We consider a scene which consists of three overlapping objects, as shown in Figure 5.25a. 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. Figure 5.25b shows the landmarks in the scene that are extracted based on the cardinal curvature points using ω=20. Although all the landmarks of the needle-nose 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 5.4. The results of mapping some of the models into the scene are shown in Figures 5.26-5.27.

The effect of noise is similarly simulated as in Section 5.2. The object contour and the landmarks of a noisy image are similarly obtained as described in Section 5.2. The image of the scene having a 0 dB SNR is shown in Figure 5.28a; the corresponding extracted landmarks are shown in Figure 5.28b. Table 5.5 summarizes the results of matching each model object of the library with the scene. Note that the sequential order of the landmarks in the scene compared to those of their respective models is totally rearranged. All the resulting matches are either incorrect or undetermined. The image with a 3 dB SNR is shown in Figure 5.29a; the corresponding extracted landmarks are shown in Figure 5.29b. The results of matching each model object of the library with the scene having a SNR = 3dB are summarized in Table 5.6. Again, the sequential order of the landmarks in the scene compared to the respective model landmarks is lost. Only the match between Mindanao and the scene has a smaller error, and the rest are either mismatched or undetermined. Images with 6 dB and 10 dB SNR along with related experimental results are shown in Figures 5.30-5.35. The experimental results are summarized in Tables 5.7 and 5.8. When the sequential order of the landmarks in the scene is not severly rearranged compared to those of the respective models, matches are correctly determined yielding small match error values.

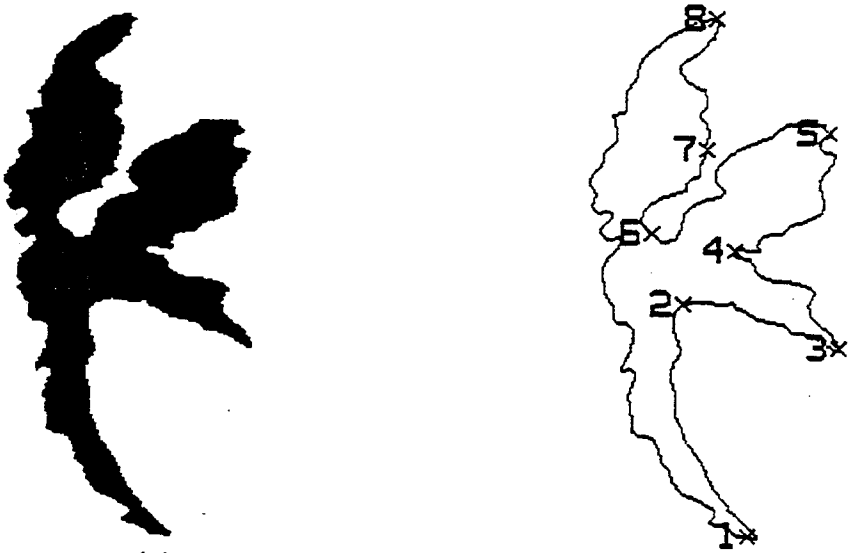
5.5. Discussion and Summary

The above experiments have demonstrated that our landmark matching task can handle occlusion reasonably well. When experimenting with the effects of noise, we have not included any preprocessing to enhance the landmark extraction task. The performance of the matching task could improve significantly if the landmarks in noisy images can be extracted more accurately. It is difficult to theoretically analyze the performance of our landmark matching method which is, in many cases, problem dependent. The performance depends on the quality of the extracted landmarks, and the number of correct landmarks in the scene that are detectable. From Chapter 4, the match error is undefined if less than three landmarks of a model are correctly matched with the scene landmarks. Therefore, when matching landmarks of a model with those of 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 above experiments, 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 the distortion in the landmark locations does not degrade the matching performance as much as the distortion in the sequential order of the landmarks.



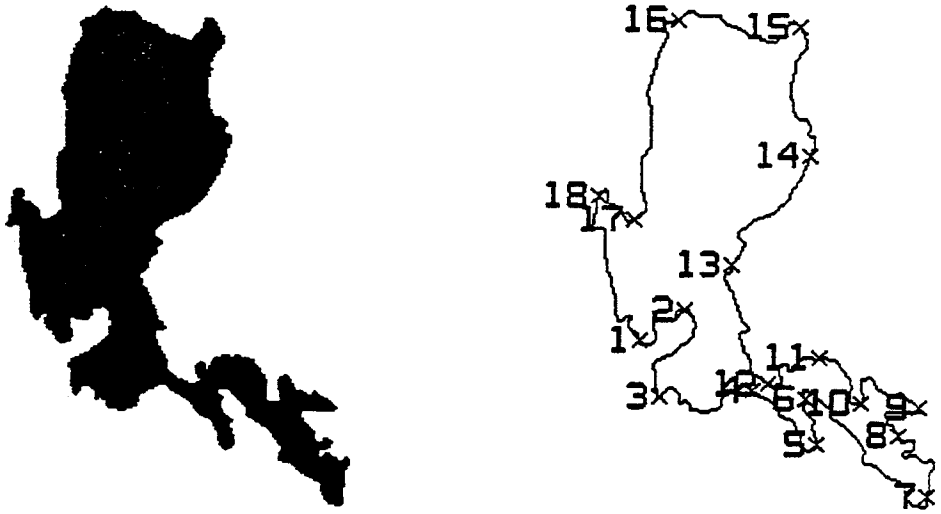
(a) (b)

Figure 5.7. The silhouette and the extracted landmarks of the island of Borneo are shown in (a) and (b), respectively. Each landmark is labeled and indicated by an "×."



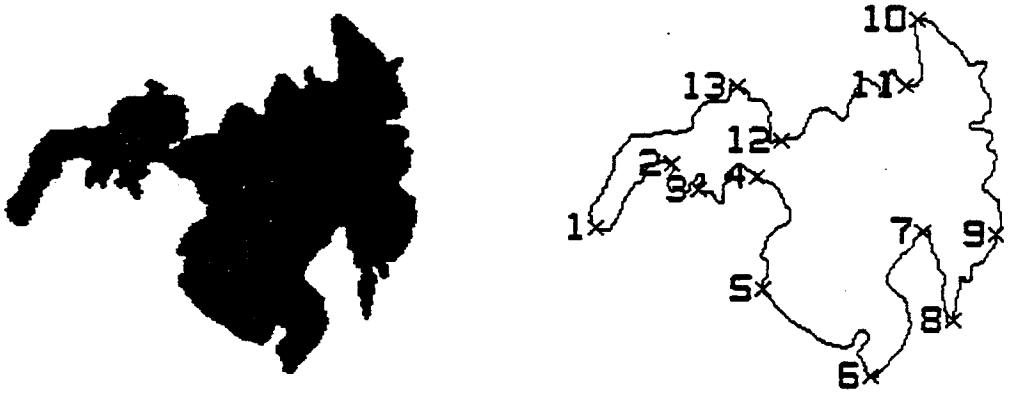
(a) (b)

Figure 5.8. The silhouette and the extracted landmarks of the island of Halmahera are shown in (a) and (b), respectively. Each landmark is labeled and indicated by an "×."



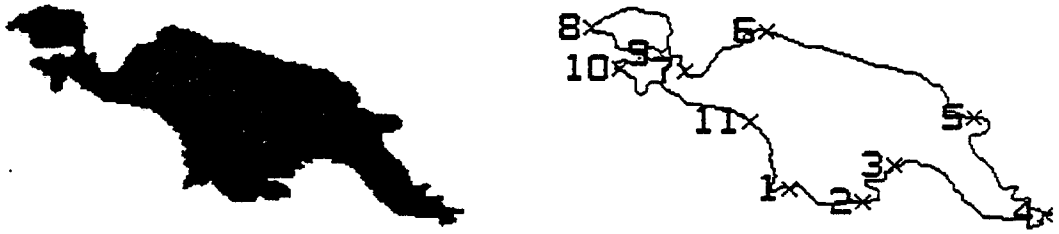
(a) (b)

Figure 5.9. The silhouette and the extracted landmarks of the island of Luzon are shown in (a) and (b), respectively. Each landmark is labeled and indicated by an "×."



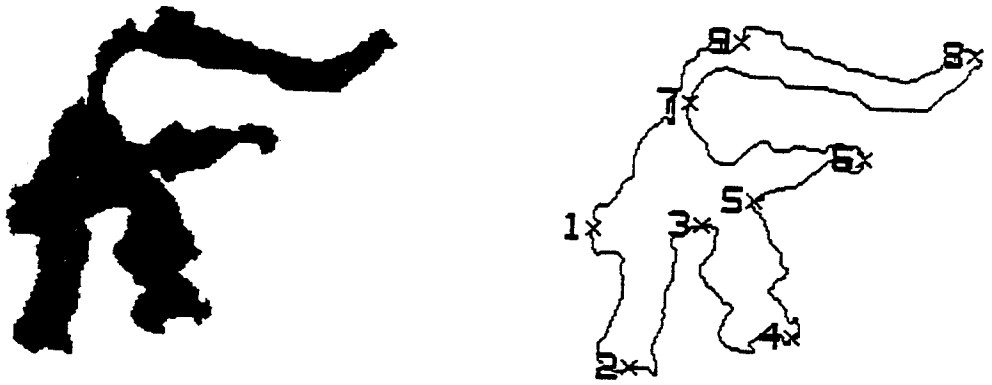
(a) (b)

Figure 5.10. The silhouette and the extracted landmarks of the island of Mindanao are shown in (a) and (b), respectively. Each landmark is labeled and indicated by an "×."



(a) (b)

Figure 5.11. The silhouette and the extracted landmarks of the island of New Guinea are shown in (a) and (b), respectively. Each landmark is labeled and indicated by an "×."



(a) (b)

Figure 5.12. The silhouette and the extracted landmarks of the island of Sulawesi are shown in (a) and (b), respectively. Each landmark is labeled and indicated by an "×."

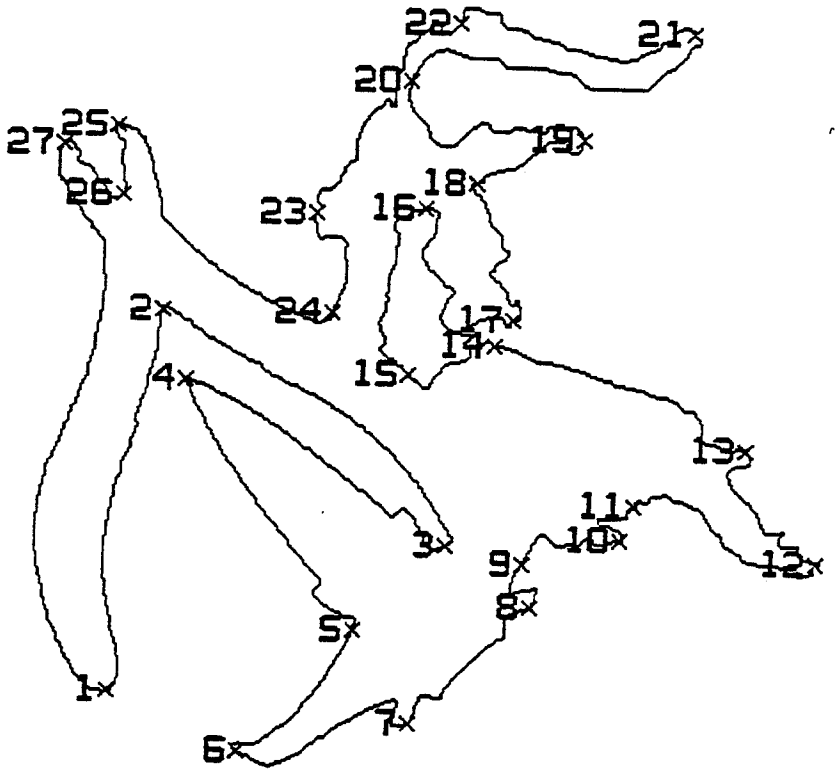


Figure 5.13. A scene which consists of four overlapping objects. Each scene landmark is labeled and indicated by an "×."

Table 5.2.
The summary of the results of matching a
library of objects with the scene shown in Figure 5.13.

Models	Model figure numbers	Total number of model landmarks	Number of the model landmarks that match with the scene	Match Error
wrench	3.35	6	2	∞
needle-nose plier	3.36	4	2	∞
wire cutter	3.34	6	5	7.56
specialty plier	3.37	6	3	53.93
wire stripper	3.38	6	6	2.49
Borneo	5.7b	7	0	∞
Halmahera	5.8b	8	4	18.22
Luzon	5.9b	18	2	∞
Mindanao	5.10b	13	3	42.94
New Guinea	5.11b	11	6	5.49
Sulawesi	5.12b	9	8	0.34
spacecraft	5.1d	7	6	0.93

Figure 5.14. The result of mapping the wire stripper into the scene shown in Figure 5.13.

Figure 5.15. The result of mapping New Guinea into the scene shown in Figure 5.13.

Figure 5.16. The result of mapping Sulawesi into the scene shown in Figure 5.13.

Figure 5.17. The result of mapping the spacecraft into the scene shown in Figure 5.13.

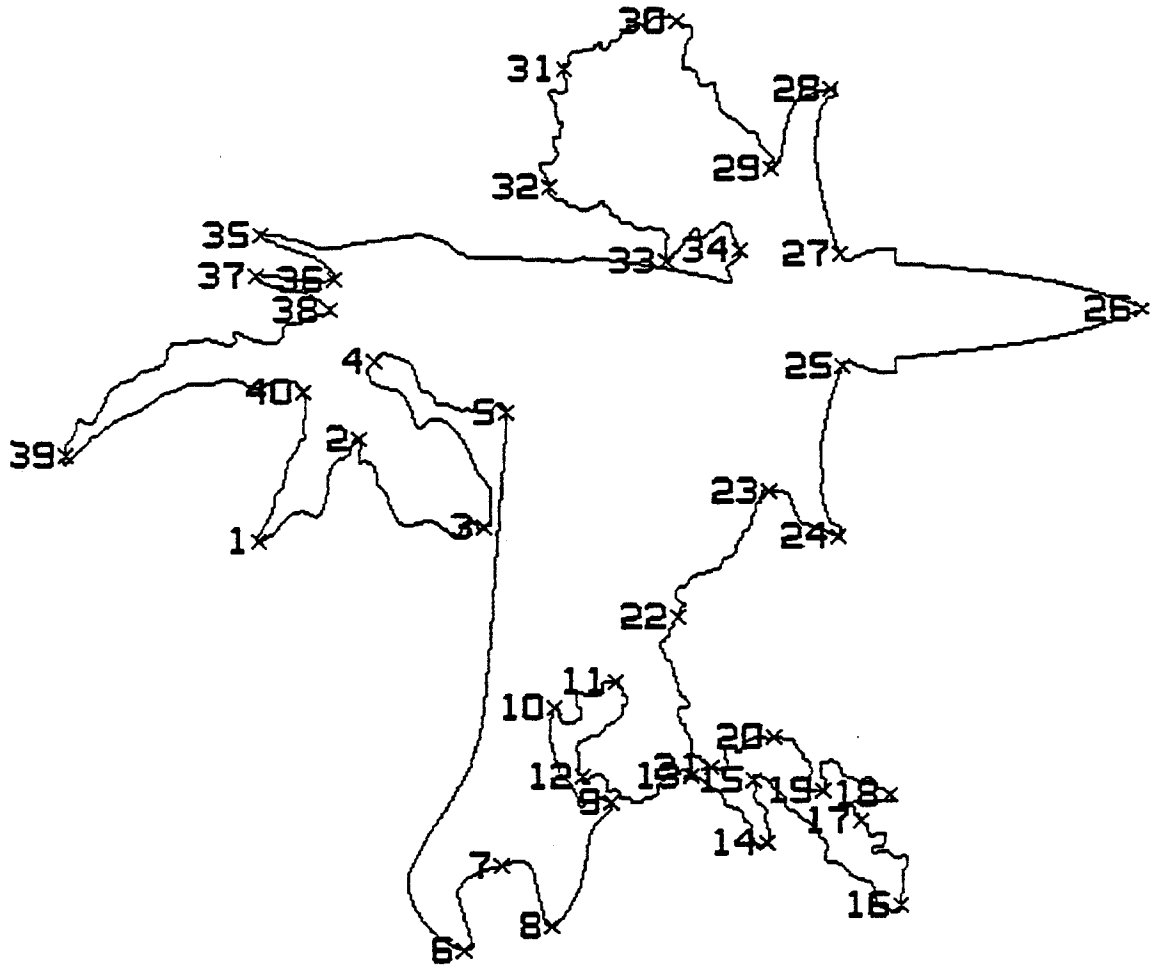


Figure 5.18. A scene which consists of six overlapping objects. Each scene landmark is labeled and indicated by an "×."

Table 5.3.
The summary of the results of matching a
library of objects with the scene shown in Figure 5.18.

Models	Model figure numbers	Total number of model landmarks	Number of the model landmarks that match with the scene	Match Error
wrench	3.35	6	4	0.74
needle-nose plier	3.36	4	0	∞
wire cutter	3.34	6	2	∞
specialty plier	3.37	6	3	7.89
wire stripper	3.38	6	2	∞
Borneo	5.7b	7	5	11.75
Halmahera	5.8b	8	6	0.57
Luzon	5.9b	18	14	0.78
Mindanao	5.10b	13	3	54.59
New Guinea	5.11b	11	4	77.81
Sulawesi	5.12b	9	4	18.08
spacecraft	5.1d	7	5	0.55

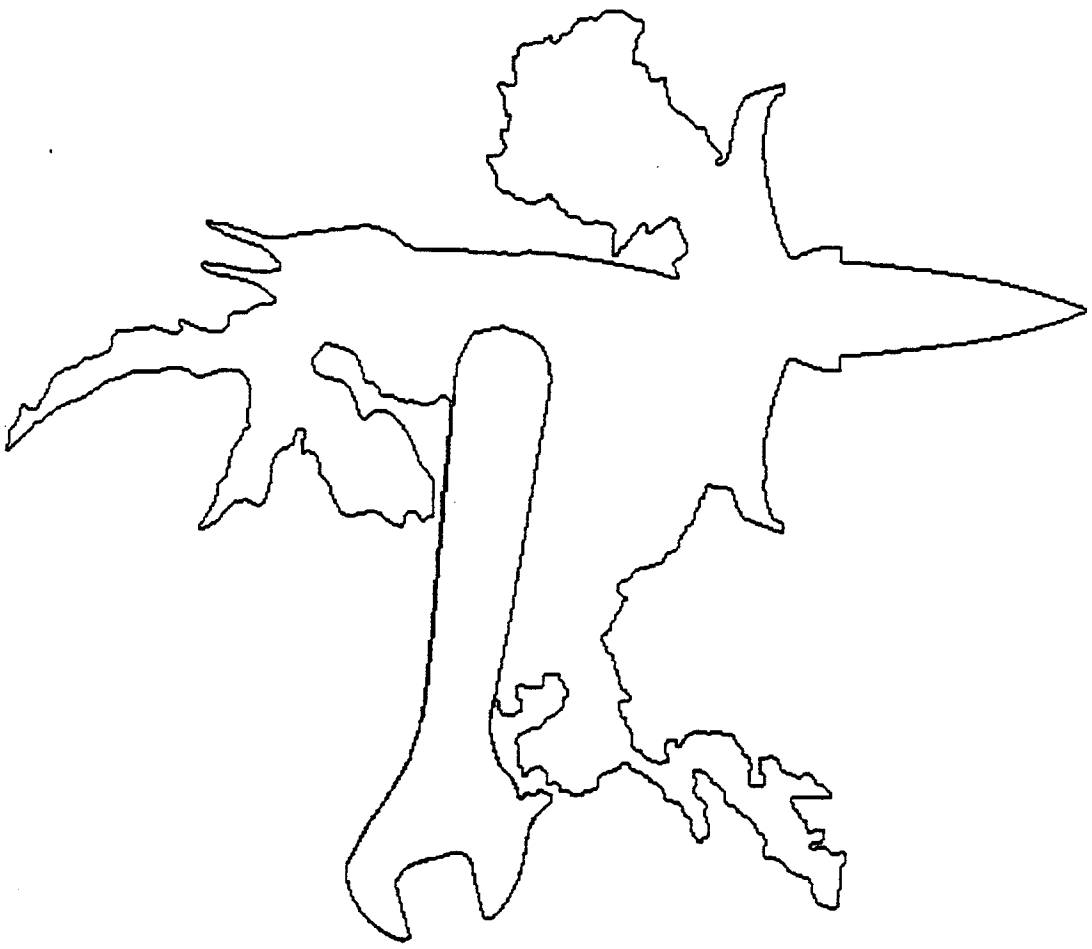


Figure 5.19. The result of mapping the wrench into the scene shown in Figure 5.18.

Figure 5.20. The result of mapping the specialty plier into the scene shown in Figure 5.18.

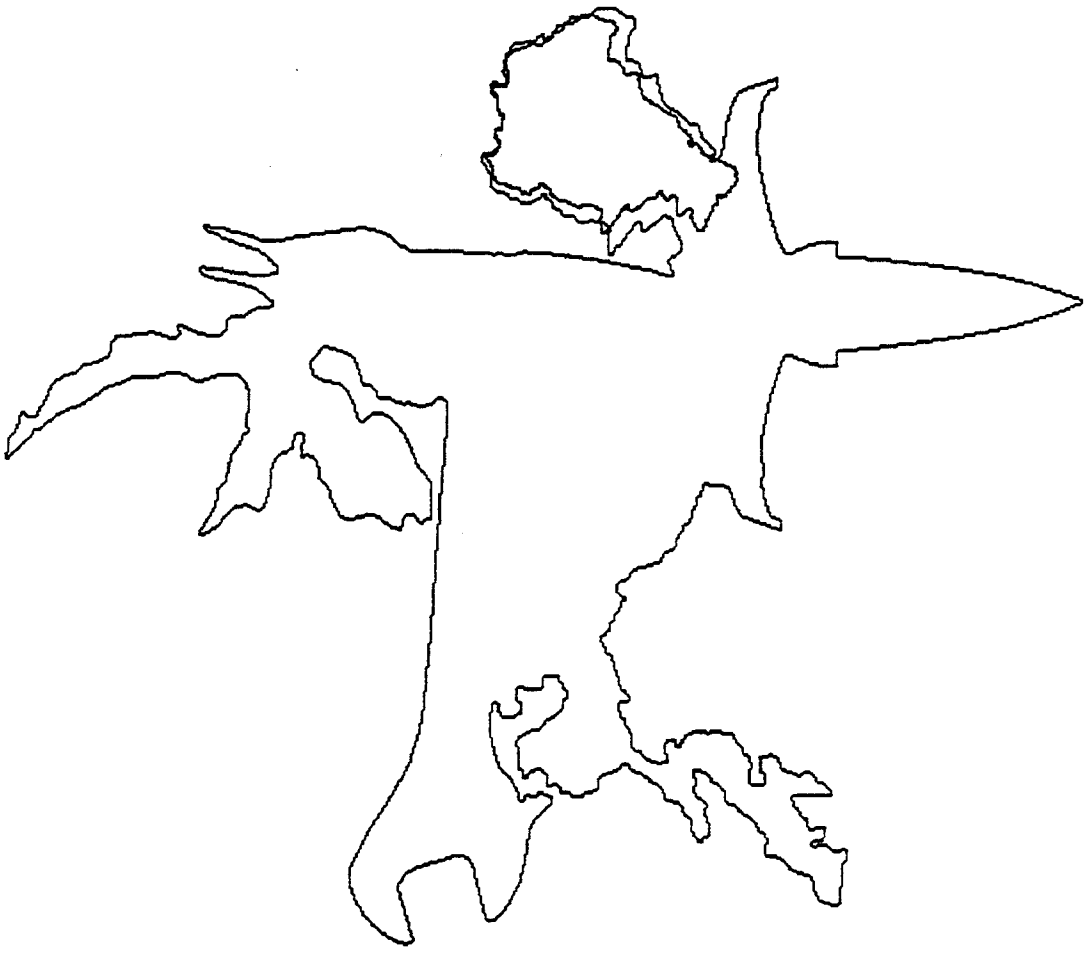


Figure 5.21. The result of mapping Borneo into the scene shown in Figure 5.18.

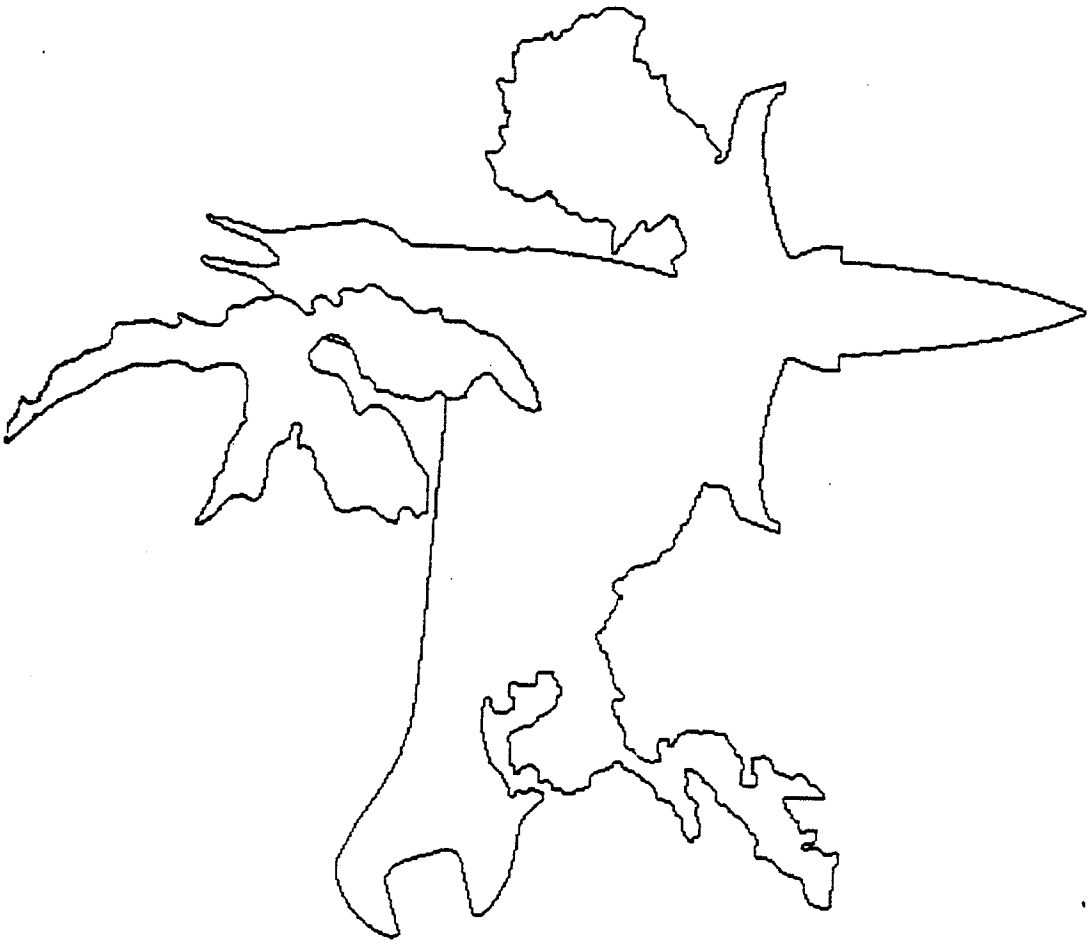


Figure 5.22. The result of mapping Halmahera into the scene shown in Figure 5.18.

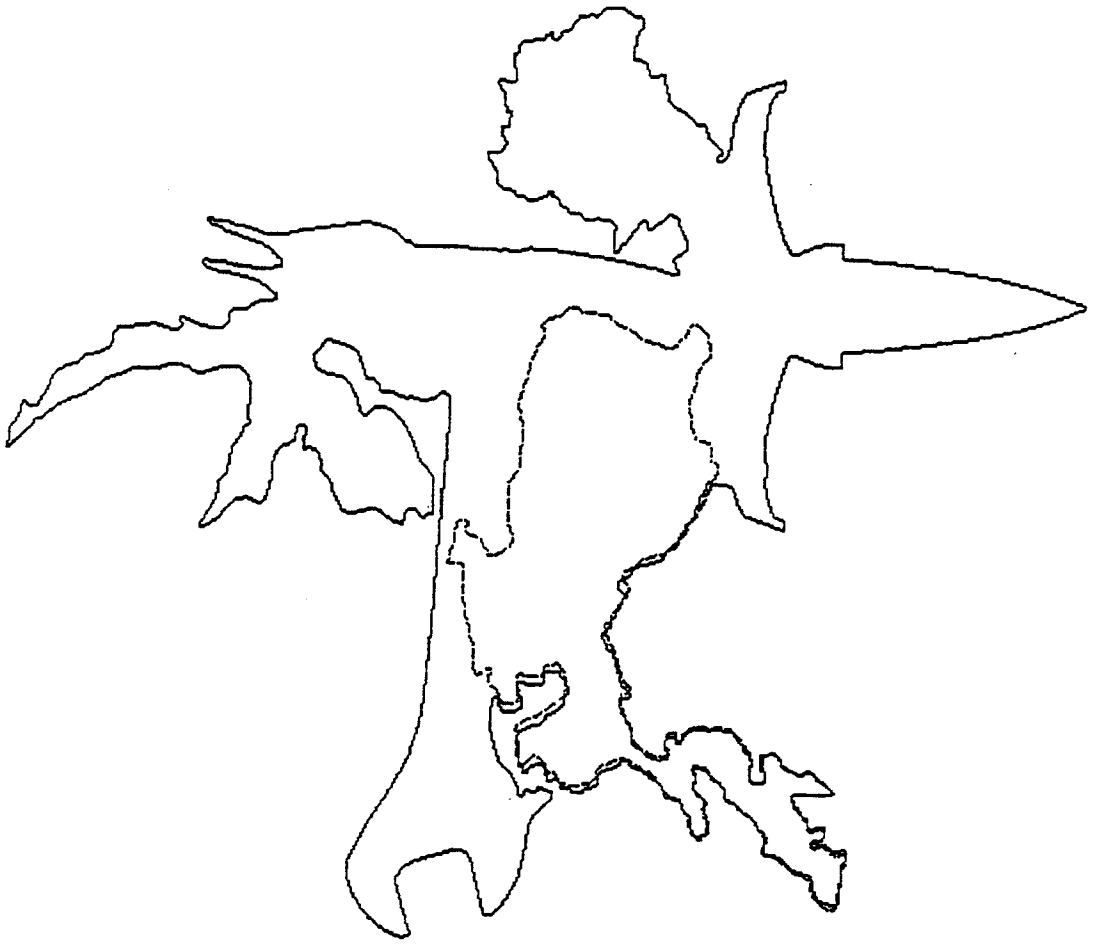


Figure 5.23. The result of mapping Luzon into the scene shown in Figure 5.18.

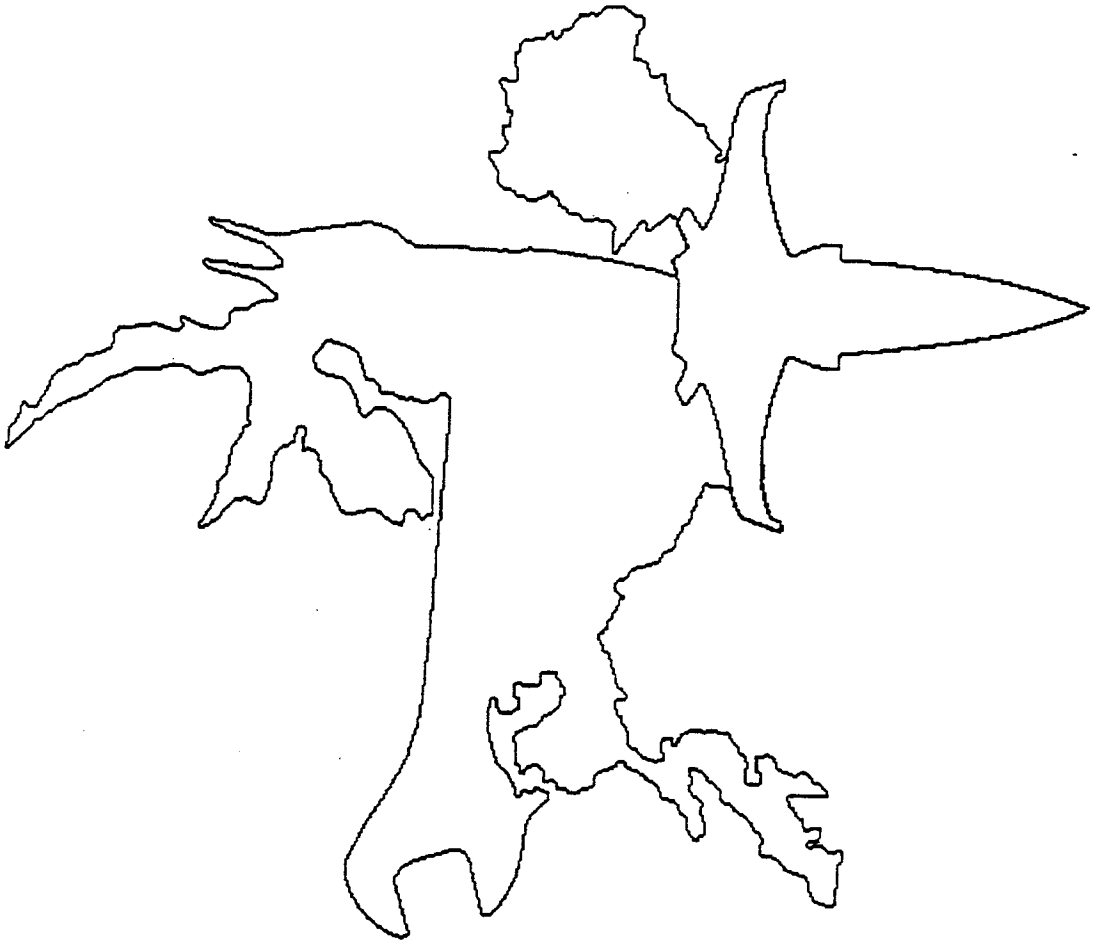


Figure 5.24. The result of mapping the spacecraft into the scene shown in Figure 5.18.

(a)

(b)

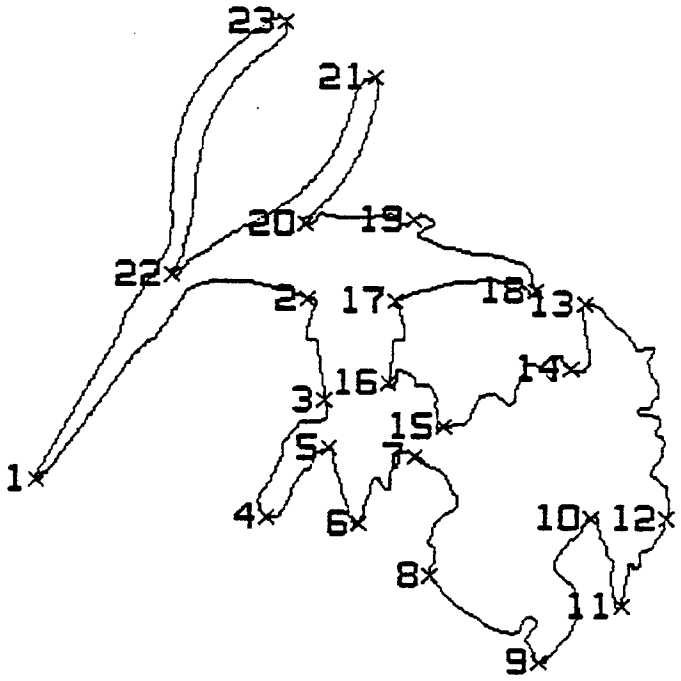


Figure 5.25. Experiments on the combined effects of noise and occlusion: noiseless case. (a) A scene which consists of three overlapping objects. (b) Extracted scene landmarks. Each landmark is labeled and indicated by an "×."

Table 5.4.
The summary of the results of matching a
library of objects with the scene shown in Figure 5.25b.

Models	Model figure numbers	Total number of model landmarks	Number of the model landmarks that match with the scene	Match Error
wrench	3.35	6	2	∞
needle-nose plier	3.36	4	4	0.24
wire cutter	3.34	6	2	∞
specialty plier	3.37	6	2	∞
wire stripper	3.38	6	4	13.96
Borneo	5.7b	7	2	∞
Halmahera	5.8b	8	2	∞
Luzon	5.9b	18	4	261.62
Mindanao	5.10b	13	10	1.40
New Guinea	5.11b	11	2	∞
Sulawesi	5.12b	9	3	140.39
spacecraft	5.1d	7	3	8.22

(a)

(b)

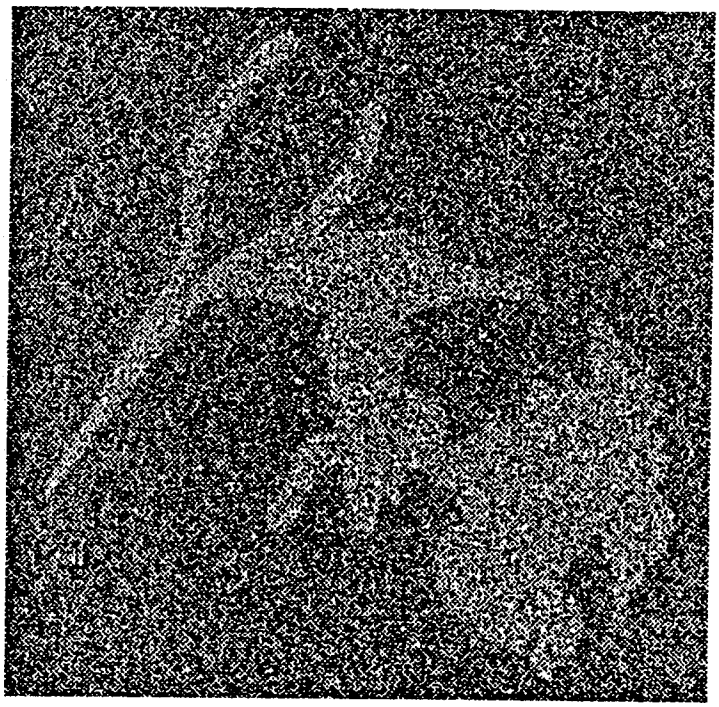
Figure 5.26. The results of mapping the needle-nose plier and Luzon into the scene (Figure 5.25b) are shown in (a) and (b), respectively.

(a)

(b)

Figure 5.27. The results of mapping Mindanao and the spacecraft into the scene (Figure 5.25b) are shown in (a) and (b), respectively.

(a)



(b)

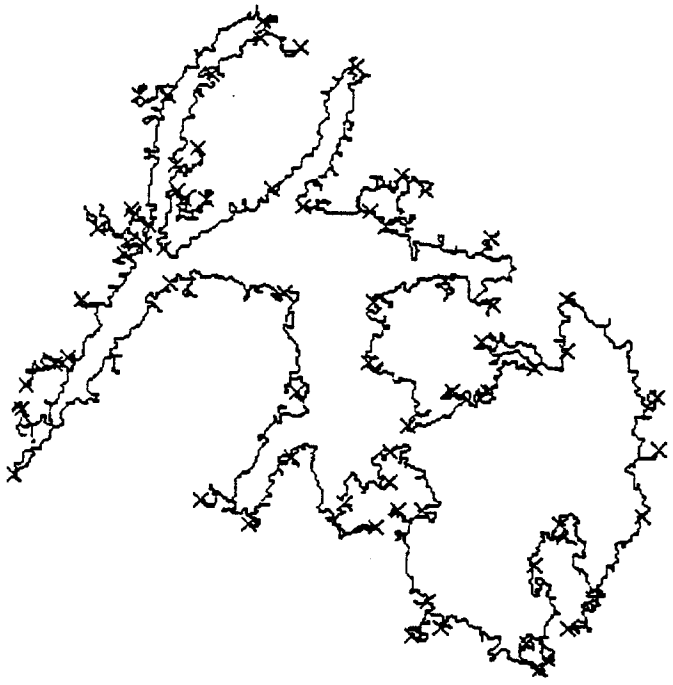


Figure 5.28. Experiments on the combined effects of noise and occlusion: SNR=0dB. (a) The noisy image of Figure 5.25a having SNR=0dB. (b) The corresponding extracted landmarks from (a). Each landmark is indicated by an "×."

Table 5.5.
The summary of the results of matching a
library of objects with the scene shown in Figure 5.28b.

Models	Model figure numbers	Total number of model landmarks	Number of the model landmarks that match with the scene	Match Error
wrench	3.35	6	0	∞
needle-nose plier	3.36	4	3	10.78
wire cutter	3.34	6	0	∞
specialty plier	3.37	6	0	∞
wire stripper	3.38	6	0	∞
Borneo	5.7b	7	3	34.28
Halmahera	5.8b	8	3	38.63
Luzon	5.9b	18	7	51.33
Mindanao	5.10b	13	6	54.32
New Guinea	5.11b	11	4	8.89
Sulawesi	5.12b	9	4	54.68
spacecraft	5.1d	7	2	∞

(a)

(b)

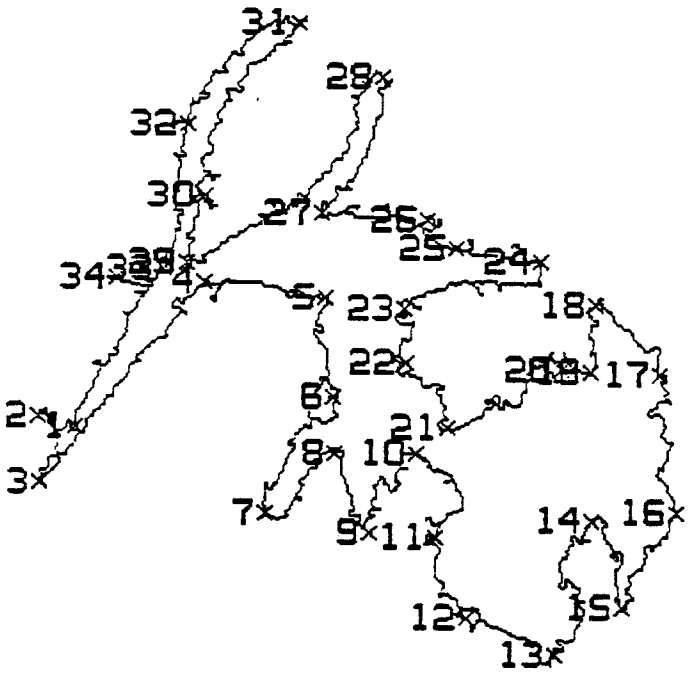
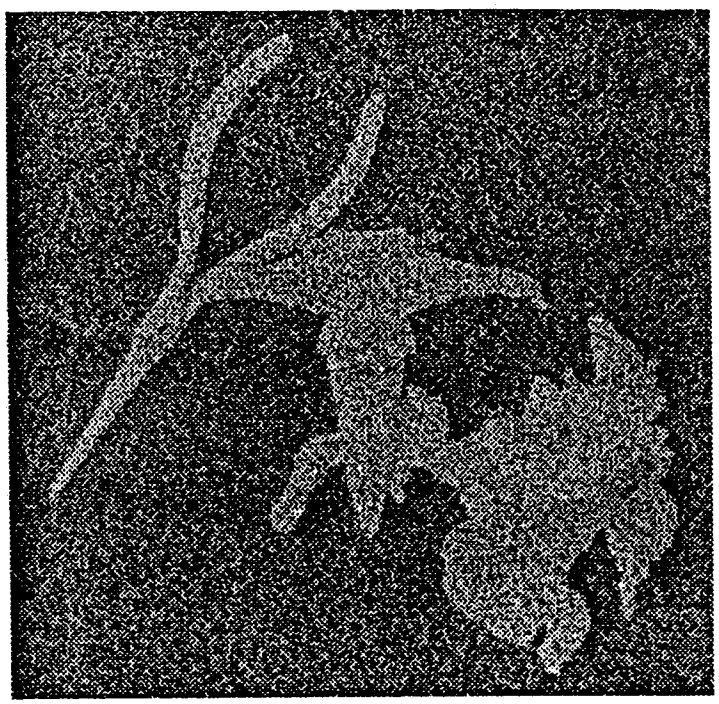


Figure 5.29. Experiments on the combined effects of noise and occlusion: SNR=3dB. (a) The noisy image of Figure 5.25a having SNR=3dB. (b) The corresponding extracted landmarks from (a). Each landmark is indicated by an "×."

Table 5.6.
The summary of the results of matching a
library of objects with the scene shown in Figure 5.29b.

Models	Model figure numbers	Total number of model landmarks	Number of the model landmarks that match with the scene	Match Error
wrench	3.35	6	2	∞
needle-nose plier	3.36	4	0	∞
wire cutter	3.34	6	0	∞
specialty plier	3.37	6	0	∞
wire stripper	3.38	6	3	78.17
Borneo	5.7b	7	3	14.42
Halmahera	5.8b	8	2	∞
Luzon	5.9b	18	4	604.01
Mindanao	5.10b	13	6	18.09
New Guinea	5.11b	11	3	116.42
Sulawesi	5.12b	9	3	71.35
spacecraft	5.1d	7	4	16.08

(a)



(b)

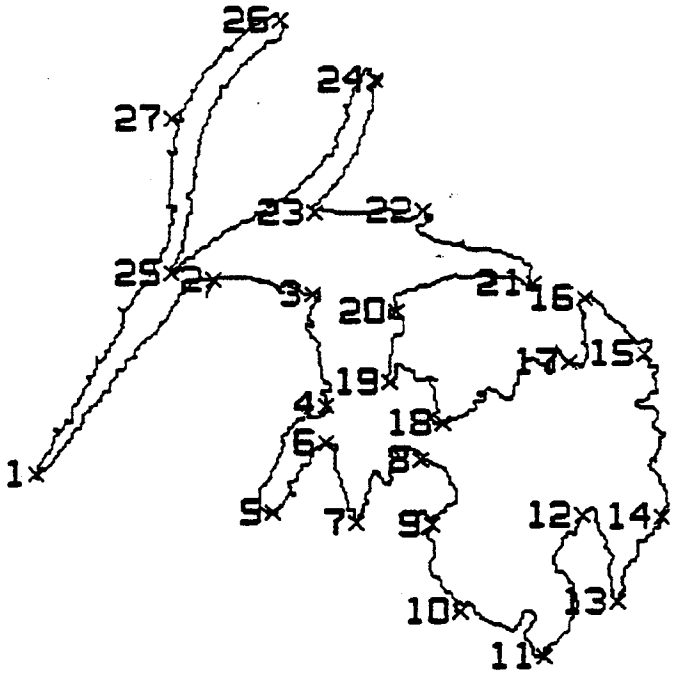
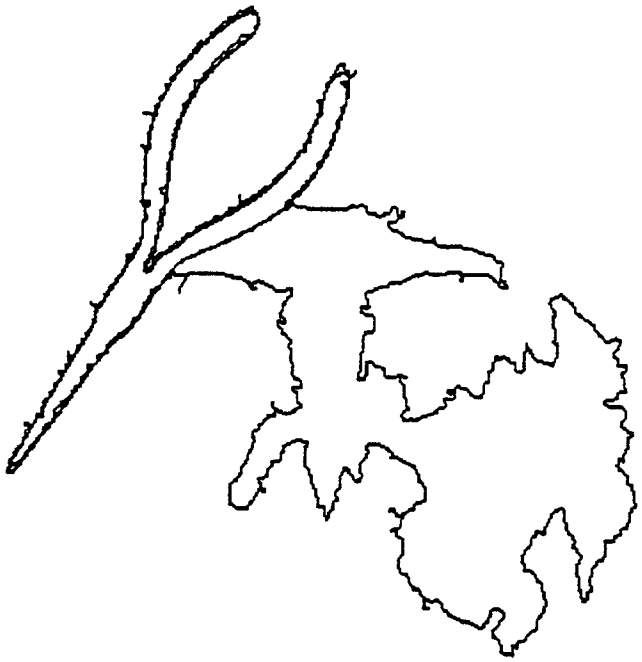


Figure 5.30. Experiments on the combined effects of noise and occlusion: SNR=6dB. (a) The noisy image of Figure 5.25a having SNR=6dB. (b) The corresponding extracted landmarks from (a). Each landmark is indicated by an "×."

Table 5.7.
The summary of the results of matching a
library of objects with the scene shown in Figure 5.30b.

Models	Model figure numbers	Total number of model landmarks	Number of the model landmarks that match with the scene	Match Error
wrench	3.35	6	0	∞
needle-nose plier	3.36	4	3	2.92
wire cutter	3.34	6	3	62.56
specialty plier	3.37	6	2	∞
wire stripper	3.38	6	4	7.75
Borneo	5.7b	7	3	121.68
Halmahera	5.8b	8	3	158.96
Luzon	5.9b	18	3	41.83
Mindanao	5.10b	13	6	14.47
New Guinea	5.11b	11	3	20.70
Sulawesi	5.12b	9	2	∞
spacecraft	5.1d	7	3	17.02

(a)



(b)

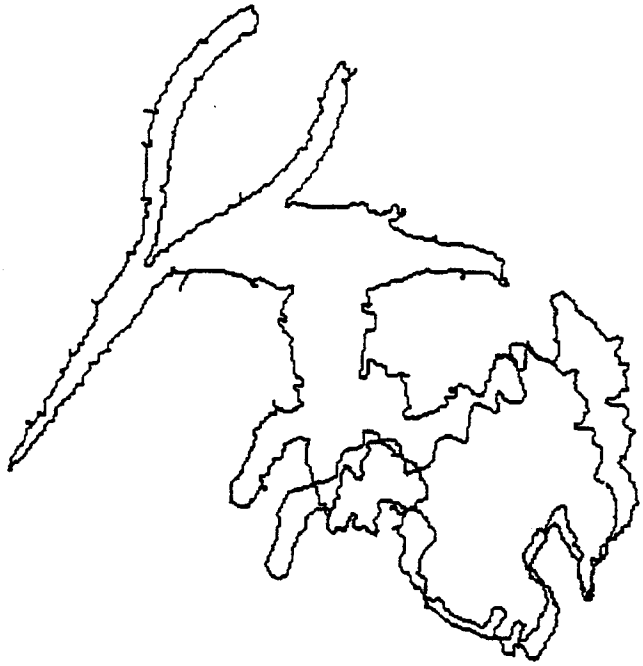


Figure 5.31. The results of mapping the needle-nose plier and Mindanao into the scene (Figure 5.30b) are shown in (a) and (b), respectively.

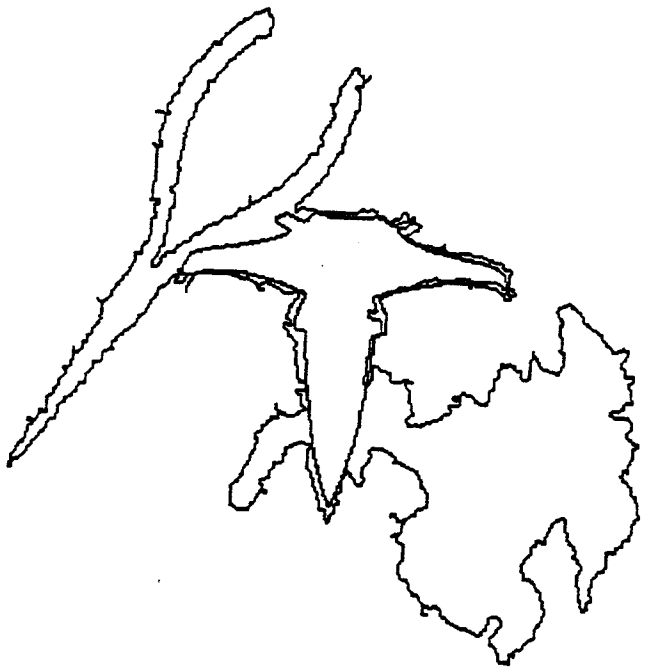


Figure 5.32. The result of mapping the spacecraft into the scene shown in Figure 5.30b.

(a)

(b)

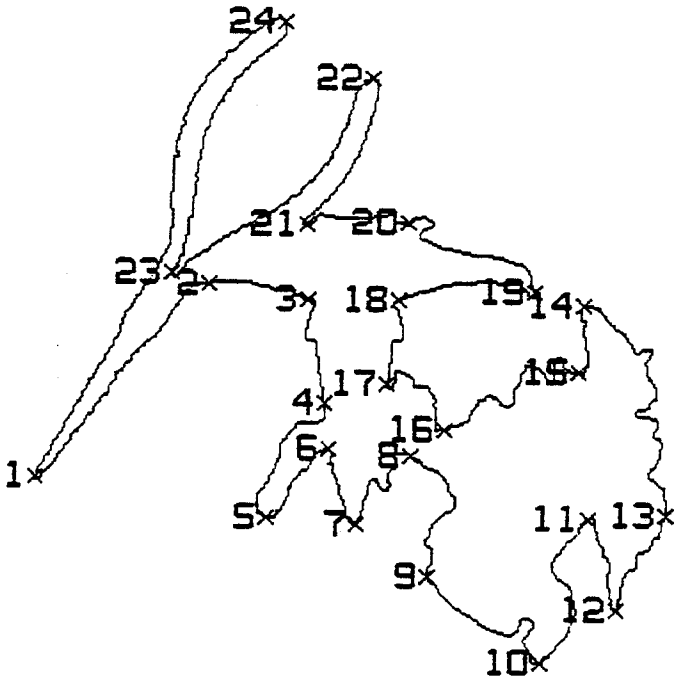


Figure 5.33. Experiments on the combined effects of noise and occlusion: SNR=10dB (a) The noisy image of Figure 5.25a having SNR=10dB. (b) The corresponding extracted landmarks from (a). Each landmark is indicated by an "×."

Table 5.8.
The summary of the results of matching a
library of objects with the scene shown in Figure 5.33b.

Models	Model figure numbers	Total number of model landmarks	Number of the model landmarks that match with the scene	Match Error
wrench	3.35	6	2	∞
needle-nose plier	3.36	4	4	0.48
wire cutter	3.34	6	2	∞
specialty plier	3.37	6	3	18.48
wire stripper	3.38	6	4	15.34
Borneo	5.7b	7	2	∞
Halmahera	5.8b	8	2	∞
Luzon	5.9b	18	4	460.50
Mindanao	5.10b	13	10	1.78
New Guinea	5.11b	11	2	∞
Sulawesi	5.12b	9	3	128.62
spacecraft	5.1d	7	3	18.18

(a)

(b)

Figure 5.34. The results of mapping the needle-nose plier and Mindanao into the scene (Figure 5.33b) are shown in (a) and (b), respectively.

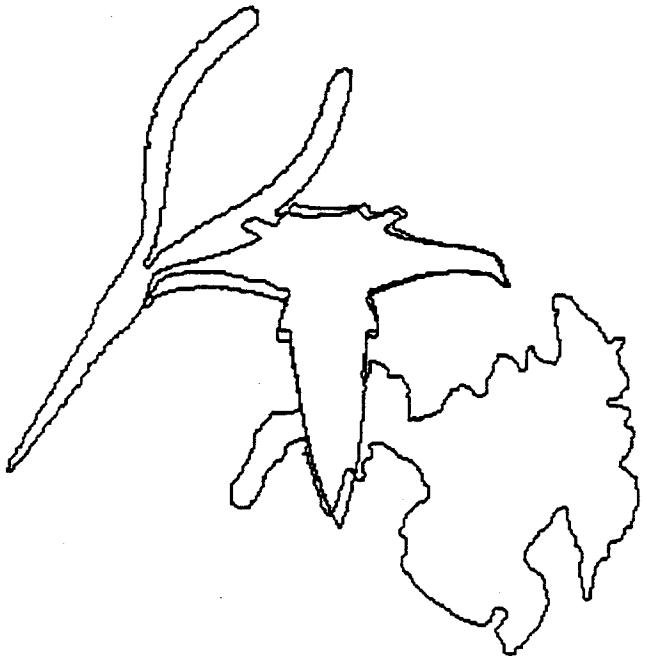


Figure 5.35. The result of mapping the spacecraft into the scene shown in Figure 5.33b.

value of 1 indicates that the two tetrahedra are similar. Recall that HDP has made use of the sequential arrangement of the landmarks to achieve the landmark matching task. If the sequential order of the landmarks in the 3-D space can be preserved, HDP can also be used to achieve the landmark matching task. When the order of the landmarks is not known, we conjecture that it is possible to construct a graph based on the sphericity values of tetrahedral transformations mapping model landmarks to scene landmarks. This graph will reflect the compatibility of the geometric structures between the model and the scene landmarks. It is worth pursuing the concept of this structured graph to achieve 3-D landmark-based shape recognition.

LIST OF REFERENCES

LIST OF REFERENCES

[Ahl66] L. V. Ahlfors, *Lectures on Quasiconformal Mappings.* Princeton, NJ: Van Nostrand, 1966.

[Att54] F. Attneave, "Some informational aspects of visual perception," *Psychological Review*, vol. 61, no. 3, pp. 183-193, 1954.

[Aya86] N. Ayache and O. D. Faugeras, "HYPER: A new approach for the recognition and positioning of two-dimensional objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 1, pp. 44-54, January 1986.

[Bar76] R. G. Bartle, *The Elements of Real Analysis.* New York: John Wiley, 1976.

[Bel65] R. Bellman and R. Kalaba, *Dynamic Programming and Modern Control Theory.* New York: Academic Press, 1965.

[Bha84] B. Bhanu and O. D. Faugeras, "Shape matching of two-dimensional objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 2, pp. 137-155, March 1984.

[Bha87] B. Bhanu and J. C. Ming, "Recognition of occluded objects: a cluster-structure algorithm," *Pattern Recognition*, vol. 20, no. 2, pp. 199-211, 1987.

[Bic77] P. J. Bickel and K. A. Doksum, *Mathematical Statistics.* San Francisco, CA: Holden-Day, 1977.

[Bol82] R. C. Bolles and R. A. Cain, "Recognizing and locating partially visible objects: the local-feature-focus method," *Int. J. of Robotics Research*, vol. 1, no. 3, pp. 57-82, Fall 1982.

[Boo78] F. L. Bookstein, *The Measurement of Biological Shape and Shape Change.* New York: Springer-Verlag, 1978.

[Boo84] F. L. Bookstein, "Tensor biometrics for changes in cranial shape," *Annals of Human Biology*, vol. 11, no. 5, pp. 413-437, 1984.

[Boo85] F. L. Bookstein, "A geometric foundation for the study of left ventricular motion: some tensor considerations," in *Digital Cardiac Imaging*, ed. A. J. Buda and E. J. Delp, pp. 65-83, Martinus Nijhoff, Dordrecht, The Netherlands, 1985.

[Bra57] L. Brand, *Vector and Tensor Analysis.* New York: John Wiley, 1957.

[Dre77] S. E. Dreyfus and A. M. Law, *The Art and Theory of Dynamic Programming.* New York: Academic Press, 1977.

[Dud77] S. A. Dudani, K. J. Breeding, and R. B. McGhee, "Aircraft identification by moment invariants," *IEEE Transactions on Computer*, vol. C-26, no. 1, pp. 39-46, January 1977.

[Fre74] H. Freeman, "Computer processing of line-drawing images," *Computing Surveys*, vol. 6, no. 1, pp. 57-97, 1974.

[Fri79] S. H. Friedberg, A. J. Insel, and L. E. Spence, *Linear Algebra.* Englewood Cliffs, NJ: Prentice-Hall, 1979.

[Gal68] R. G. Gallager, *Information Theory and Reliable Communication.* New York: Wiley, 1968.

[Gan69] D. Gans, *Transformations and Geometries.* New York: Appleton-Century-Crofts, 1969.

[Gor88] J. W. Gorman, O. R. Mitchell, and F. P. Kuhl, "Partial shape recognition using dynamic programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-10, no. 2, pp. 257-266, March 1988.

[Gra61] F. A. Graybill, *An Introduction to Linear Statistical Models.* New York: McGraw-Hill, 1961, Volume 1 .

[Gug63] H. W. Guggenheimer, *Differential Geometry.* New York: McGraw-Hill, 1963.

[Hod55] J. L. Hodges, Jr., "On the noncentral beta-distribution," *The Annals of Mathematical Statistics*, vol. 26, no. 4, pp. 648-653, 1955.

[Hu62] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. IT-8, no. 1, pp. 179-187, February 1962.

[Joh70] N. L. Johnson and S. Kotz, *Continuous Univariate Distributions - 2.* New York: John Wiley, 1970.

[Kap81] W. Kaplan, *Advanced Mathematics for Engineers.* Reading, MA: Addison-Wesley, 1981.

[Kno86] T. F. Knoll and R. C. Jain, "Recognizing partially visible objects using feature indexed hypotheses," *IEEE Transactions on Robotics and Automation*, vol. RA-2, no. 1, pp. 3-13, March 1986.

[Koc87] M. W. Koch and R. L. Kashyap, "Using polygons to recognize and locate partially occluded objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 4, pp. 483-494, July 1987.

[Lev85] M. D. Levine, *Vision in Man and Machine.* New York: McGraw-Hill, 1985.

[Mok86] F. Mokhtarian and A. Mackworth, "Scale-based description and recognition of planar curves and two-dimensional shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 1, pp. 34-43, January 1986.

[Mui82] R. J. Muirhead, *Aspects of Multivariate Statistical Theory.* New York: John Wiley, 1982.

[O'N66] B. O'Neill, *Elementary Differential Geometry.* New York: Academic Press, 1966.

[Ott88] P. J. van Otterloo, *A Contour-Oriented Approach to Digital Shape Analysis.* Den Haag, The Netherlands: Cip-Data Koninklijke Bibliotheek, 1988.

[Pav74] T. Pavlidis and S. L. Horowitz, "Segmentation of plane curves," *IEEE Transactions on Computers*, vol. C-23, no. 8, pp. 860-870, 1974.

[Pav78] T. Pavlidis, "A review of algorithms for shape analysis," *Computer Graphics and Image Processing*, vol. 7, pp. 243-258, April 1978.

[Pav80] T. Pavlidis, "Algorithms for shape analysis of contours and waveforms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-2, no. 4, pp. 301-312, July 1980.

[Per77] E. Persoon and K. S. Fu, "Shape discrimination using Fourier descriptors," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-7, no. 2, pp. 170-179, March 1977.

[Pri84] K. E. Price, "Matching closed contours," in *Seventh International Conference on Pattern Recognition*, pp. 990-992, Montreal, Canada, July 30 - Aug 2, 1984.

[Ram72] U. Ramer, "An iterative procedure for the polygonal approximation of plane closed curves," *Computer Graphics and Image Processing*, vol. 1, no. 3, pp. 244-256, November 1972.

[Seb63] G. A. F. Seber, "The non-central chi-squared and beta distributions," *Biometrika*, vol. 50, pp. 542-544, 1963.

[Seb77] G. A. F. Seber, *The non-central chi-squared and beta distributions.* New York: John Wiley, 1977.

[Sko86] D. J. Skorton, S. M. Collins, and R. E. Kerber, "Digital image processing and analysis in echocardiography," in *Cardiac Imaging and Image Processing*, ed. S. M. Collins and D. J. Skorton, pp. 171-205, McGraw-Hill, New York, 1986.

[Sok51] I. S. Sokolnikoff, *Tensor Analysis.* New York: John Wiley, 1951.

[Tho72] G. B. Thomas, Jr., *Calculus and Analytic Geometry.* Reading, MA: Addison-Wesley, 1972.

[Tor86] V. Torre and T. A. Poggio, "On edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 2, pp. 147-163, March 1986.

[Tur85] J. L. Turney, T. N. Mudge, and R. A. Volz, "Recognizing partially occluded parts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-7, no. 4, pp. 410-421, July 1985.

[Vai61] J. Vaisala, "On quasiconformal mappings in space," *Annales Academiae Scientiarum Fennicae*, no. A.I. 298, pp. 1-36, 1961.

APPENDIX

APPENDIX

A DETAILED PROOF OF LEMMA 2.36

Lemma 2.36 If W is a non-central Beta random variable, with 2, 2 degrees of freedom, and the noncentrality, ρ, that is, $f_W(w) = \beta(w;1,1,\rho)$, then

$$E(W) = 1 - \frac{1}{\rho} + \frac{1}{\rho^2} - \frac{1}{\rho^2}e^{-\rho}$$

$$E(W^2) = 1 - \frac{2}{\rho} + \frac{4}{\rho^2}(1 - \frac{1}{\rho} + \frac{e^{-\rho}}{\rho})$$

$$Var(W) = \frac{1}{\rho^2}(1 - \frac{2}{\rho} - \frac{1}{\rho^2} + 2e^{-\rho}(1 + \frac{1}{\rho} + \frac{1}{\rho^2} + \frac{e^{-\rho}}{\rho^2})) .$$

Proof: By making use of Theorem 2.33, Lemma 2.35, and the moments of a Poisson random variable, we have

$$E(W) = \int_0^1 w f_W(w)\, dw$$

$$= \int_0^1 w\ \beta(w;1,1,\rho)\, dw$$

By Theorem 2.33,

$$= \int_0^1 w \left(\sum_{j=0}^{\infty} e^{-\rho}\frac{\rho^j}{j!}\beta(w;1+j,1) \right) dw$$

$$= \sum_{j=0}^{\infty} \left(e^{-\rho}\frac{\rho^j}{j!}\int_0^1 w\,\beta(w;1+j,1)dw \right)$$

By Lemma 2.35,

$$= \sum_{j=0}^{\infty} \left(e^{-\rho} \frac{\rho^j}{j!} \frac{1+j}{2+j} \right)$$

$$= \sum_{j=0}^{\infty} \left((j+1)^2 \frac{e^{-\rho}\rho^j}{(j+2)!} \right)$$

Let $i=j+2$,

$$= \sum_{i=2}^{\infty} \left((i-1)^2 \frac{e^{-\rho}\rho^{(i-2)}}{i!} \right)$$

$$= \frac{1}{\rho^2} \left\{ \sum_{i=0}^{\infty} \left((i^2-2i+1) \frac{e^{-\rho}\rho^i}{i!} \right) - e^{-\rho} \right\}$$

Using the moments of a Poisson random variable with parameter ρ,

$$E(W) = 1 - \frac{1}{\rho} + \frac{1}{\rho^2} - \frac{1}{\rho^2} e^{-\rho} \quad . \tag{A.1}$$

Note that moments of a Poisson random variable can be obtained by using the moment generating function of the Poisson random variable.

Similarly,

$$E(W^2) = \sum_{j=0}^{\infty} \left(e^{-\rho} \frac{\rho^j}{j!} \int_0^1 w^2 \beta(w;1+j,1) dw \right)$$

$$= \sum_{j=0}^{\infty} \left(e^{-\rho} \frac{\rho^j}{j!} \frac{1+j}{j+3} \right)$$

$$= \frac{1}{\rho^3} \sum_{j=0}^{\infty} \left((j+1)^2 (j+2) e^{-\rho} \frac{\rho^j}{(j+3)!} \right)$$

$$= \frac{1}{\rho^3} \sum_{i=3}^{\infty} \left((i-2)^2 (i-1) e^{-\rho} \frac{\rho^i}{i!} \right)$$

$$= \frac{1}{\rho^3}\left\{\sum_{i=0}^{\infty}\left((i^3-5i^2+8i-4)e^{-\rho}\frac{\rho^i}{i!}\right)+4e^{-\rho}\right\}$$

$$= 1-\frac{2}{\rho}+\frac{4}{\rho^2}(1-\frac{1}{\rho}+\frac{e^{-\rho}}{\rho}) \quad . \tag{A.2}$$

The variance is obtained by using Equations A.1 and A.2:

$$Var(W) = E(W^2) - (E(W))^2$$

$$= \frac{1}{\rho^2}(1-\frac{2}{\rho}-\frac{1}{\rho^2}+2e^{-\rho}(1+\frac{1}{\rho}+\frac{1}{\rho^2}+\frac{e^{-\rho}}{\rho^2})) \quad . \qquad \square$$

VITA

VITA

Nirwan Ansari was born in Jambi, Indonesia on November 17, 1958, to his parents, Sugiarto and Linda. He first attended school at the age of 10 at Hong Kong. He received the BSEE (*summa cum laude*) from the New Jersey Institute of Technology in 1982, and the MSEE from the University of Michigan in 1983. He expects to receive his doctoral degree from Purdue University in August 1988. He has accepted a faculty position with the Department of Electrical Engineering at the New Jersey Institute of Technology starting Fall 1988. His current research interests include image processing, computer vision, signal processing, and communication theory. He is a member of Eta Kappa Nu, Tau Beta Pi, and the IEEE.