12-1-1994

Recognition and Localization of a Flat Symmetric Object from a Single Perspective Image

Andoo P. Yang
Purdue University School of Electrical Engineering

David M. Chelberg
Purdue University School of Electrical Engineering

Recognition and Localization of a Flat Symmetric Object from a Single Perspective Image

Andoo P. Yang
David M. Chelberg

TR-EE 94-40
December 1994

School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907-1285

Recognition and Localization of a Flat Symmetric Object from a Single Perspective Image

Andoo P. Yang and David M. Chelberg

School of Electrical Engineering

1285 Electrical Engineering Building

Purdue University

West Lafayette, Indiana 47907-1285

ABSTRACT

A flat symmetric object viewed under perspective (orthographic) projection yields a convergent (skew) symmetry. Unlike skew symmetry where the axes constrain the orientation of the underlying object to lie on a hyperbola in the gradient space, for convergent symmetry the object orientation can be uniquely determined from the axis and convergent point except for some special cases. However, detection of the axis and convergent point of a convergent symmetry has not yet received much attention. We present a technique which can detect the axis and convergent point of a convergent symmetric contour in an image using distinguished points such as bitangents, corners, and inflection points on the contour. In addition to recovering the orientation of an object directly from the axis and convergent point of its convergent symmetry, when a model database is available, our technique is also able to recognize and locate the object by using the cross ratio as an indexing function. The orientation information is also used for verification purposes. Our technique is simple, efficient, and accurate. It is also robust to noise and partial occlusion. Experimental results on real images are shown.

I Introduction

Since many man-made and natural objects have flat bilaterally symmetric contours, symmetric features can be useful cues for recovering the 3-dimensional shape of an object from its image. In addition, symmetric features can be used as indexing functions for model-based object recognition. However, when symmetric figures are projected (orthographically or perspectively) onto an image, they do not remain symmetric. Therefore, recovery of symmetric features in a projected image is an interesting problem which has been investigated by many researchers.

The concept of a skewed symmetry, obtained from a flat symmetric object viewed from an oblique direction assuming orthographic projection, was introduced by Kanade [1]. Unlike traditionally defined symmetry where the symmetric axis and transverse axis are perpendicular to each other, skewed symmetry relaxes this condition a little, i.e., the skewed transverse axis is not necessarily perpendicular to the skewed symmetric axis, but at a fixed angle to it (see Figure 1). Kanade showed that the parameters (axes) of skewed symmetry constrain the orientation of an underlying object to lie on a hyperbola in gradient space and successfully recovered the 3-D shape of some objects from a single image by combining this constraint with other heuristics. Since then, many researchers have reported different techniques for finding axes of skewed symmetry [2, 3, 4, 5] and for resolving the remaining gradient ambiguity in order to recover the orientation of the object uniquely [6, 7, 8].

A flat symmetric object, when viewed under perspective projection, yields a convergent symmetry. This concept was introduced by Ulupinar and Nevatia [9] based on the fact that the perspective projection of any set of parallel lines that are not parallel to the image plane converge to a point, called the convergent (vanishing) point (see Figure 2). They showed that by using the axis and convergent point of a convergent symmetry, one can uniquely determine the orientation of an object (except for some special cases). However, practical methods to find the axis and convergent point of a convergent symmetry were not described in their paper.

Several methods for detection of vanishing points have been reported [10, 11, 12, 13]. Most of these methods are based on line segments extracted from the images; vanishing points are

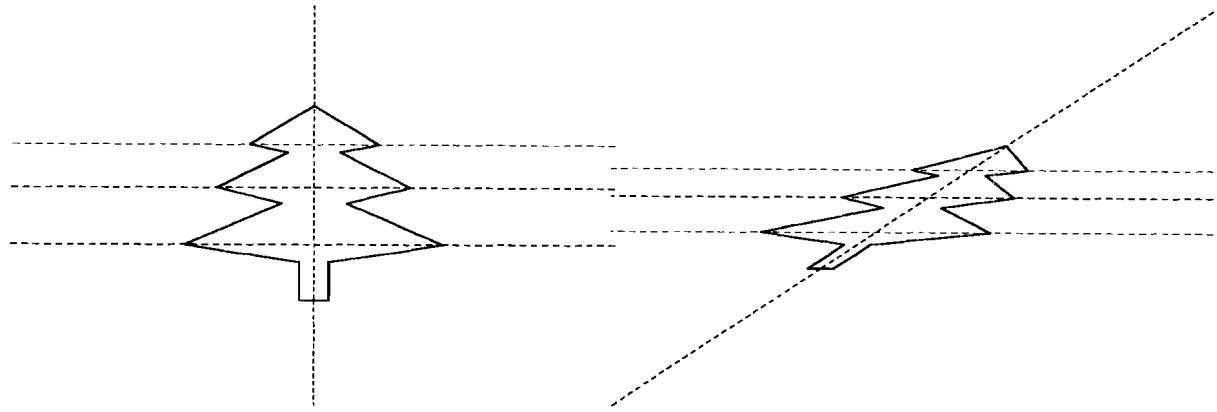


Figure 1: A symmetrical tree pattern and its skewed symmetry.

then determined by finding the intersection points of the supporting lines of the extracted line segments. These methods do not work for curved objects and are substantially different from the method for detecting convergent (vanishing) points described in this paper.

Glachet et al. [14] recently developed a method for detecting a convergent symmetry. They used a prediction-verification scheme based on the detection of straight lines on the convex hull of an extracted closed contour from the image to find the axis and convergent point. They also pointed out that the special case stated in Ulupinar and Nevatia [9], in which the orientation of an object cannot be determined from the convergent symmetry, can be extended. They found that as soon as the plane passing through the optical center and the axis of convergent symmetry is orthogonal to the lines of symmetry of the object, the orientation of the object cannot be determined from the convergent symmetry. One assumption they made in trying to find the location of an object is that the height of the object is known. This is not usually true in practice. For instance, they used this method in their work on recovering the scaling function of a straight homogeneous generalized cylinder (SHGC) from a single perspective view [15]. In their work, they assumed that the base of a SHGC which is symmetric is visible, and claimed that the localization of the SHGC can he made by using this method. However, it is not clear how they can automatically solve the localization problem without knowing the dimension of the base.

After invariant properties of SHGCs were reported by Ponce, Chelberg, and Mann [16], many

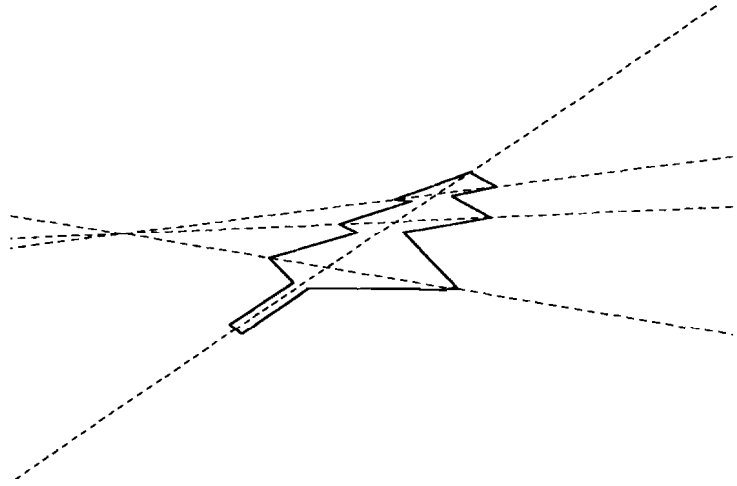


Figure 2: The convergent symmetry of the tree in last figure.

methods for detecting the projection of the axis of SHGC have been developed [17, 18, 19]. Liu et al. [19] described a technique for finding the projection of the axis of rotationally symmetric surfaces. The cross-ratio of four collinear points is then used as an indexing function to recognize those surfaces. They also showed that the recognition system can be extended to recognize Straight Homogeneous Generalized Cylinders. However, they claimed that it is extremely hard to arrange the camera such that lines joining corresponding pairs of distinguished points are more than a few degrees away from parallel, and used a parallel constraint in determining corresponding bitangents. This explicitly exclctdes the case of convergent symmetry. By replacing the parallel constraint with an ordering constraint which will be described later, we are able to find the axis and convergent point of convergent symmetries and recover the orientation of the object. Using the cross-ratio as an indexing function together with the orientation information, we can recognize flat symmetric objects efficiently and accurately.

The contributions of this paper are as follows: (1) A new robust method for detecting convergent symmetries is proposed. With this method, the orientation of an observed flat symmetric object can be recovered from a single perspective image; (2) An efficient and accurate object recognition process using the cross ratio as an indexing function and orientation information

for verification has been developed. When a model database is available, an observed flat symmetric object can be recognized and its location can be uniquely determined.

This paper is organized as follows: in section 2 we define the notations, terms, and properties used in this paper. In section **3** we describe the technique for finding the axis and convergent point of a convergent symmetry. In section 4 we show how a flat symmetric object can be recognized and located from a single perspective image using cross-ratio and orientation information. In section 5, some experimental results for real images taken by a camera are shown.

II Notations, terms, and properties

In this section we introduce the notations, terms, and properties which are used in this paper. Most of the properties here have been reported in the literature and are indicated accordingly.

A Perspective Projection

Throughout this paper we use a viewer (camera) centered coordinate system with the x-axis to the right, the y-axis upward, and the z-axis pointing toward the viewer (i.e., a right-handed coordinate system). The optical center of the camera (O) is at the origin. With this convention, the z components of the coordinates of points in front of the camera are negative. The image plane is parallel to the xy-plane at a distance f from the origin along the negative z-axis (i.e., $z = -\mathrm{f}$). Therefore, an object point $(\mathrm{x}, \mathrm{y}, \mathrm{z})$ is perspectively projected onto the image point (u,v), where $\mathrm{u} = -x\mathrm{f}/z$ and $\mathrm{v} = -y\mathrm{f}/z$, which is the point at which a line through the origin and $(\mathrm{x}, \mathrm{y}, \mathrm{z})$ intersects the image plane. Note that any image point (u,v) is also a point $(\mathrm{u}, v, -\mathrm{f})$ in the x-y-z coordinate system (see Figure **3).**

Some properties of perspective projection are used in this paper.

Property 1. Let S be a plane having plane normal $(p, \mathrm{q}, 1)$ and intersecting the z-axis at $z = \mathrm{D}$ (i.e., the plane is defined by $\mathrm{px} + \mathrm{qy} + z - \mathrm{D} = 0$). Let $\mathrm{P} = (\mathrm{u}, \mathrm{v})$ be a point in the image; it must correspond to some point Q on plane S. Suppose the image of Q is P, then $\mathrm{Q} = (au, av, -af)$ for some value a. Since Q lies on S, $p(au) + q(av) + (-\mathrm{af}) - \mathrm{D} = 0$.

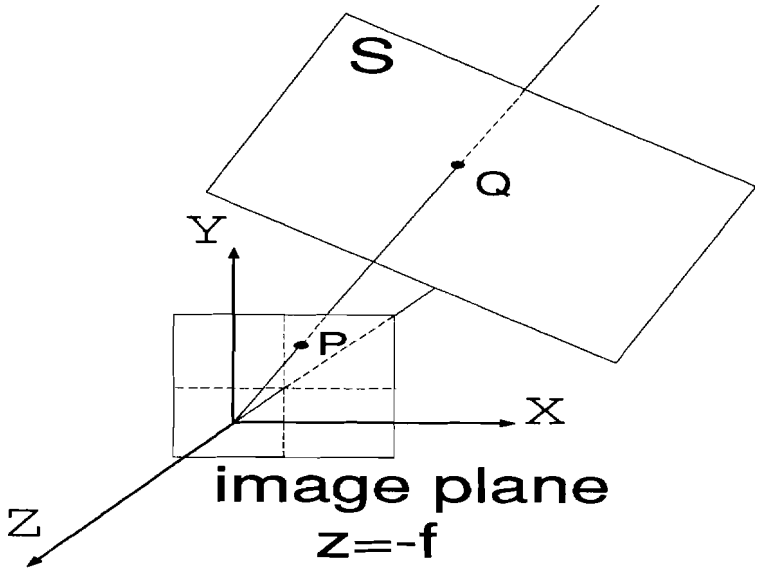


Figure 3: Perspective projection model.

Solving this equation for a yields a $= D/(pu + qv - f)$ and

$$Q = \frac{D}{pu + qv - f}(u, v, -f). \tag{1}$$

Q is sometimes called the back-projection of image point P onto plane S [20]. We will use this property to project some pairs of corresponding distinguished points on the image plane back to 3-D space and check whether the corresponding projected points are symmetric with respect to some back projected axis in 3-D. A measure of symmetry will be described in section III. Since we are only interested in the measure of symmetry, once the plane normal of S is known, the value D can be selected arbitrarily. That is, we do not have to know the depth of an object in order to compute the measure of symmetry.

Property 2. Distinguished points, such as corners, inflection points, and bitangents, are invariant under perspective projections and can be located before and after projection (see Mundy et al. [21]). These distinguished points will be used as invariant descriptors in our approach for finding the axis and convergent point of a convergent symmetry.

Property 3. Let (p_1, p_2, p_3, p_4) be four different collinear points. The cross-ratio of the four

points is defined by:

$$\tau(p_1, p_2, p_3, p_4) = \frac{D(p_1p_3)D(p_2p_4)}{D(p_1p_4)D(p_2p_3)} \tag{2}$$

where $D(p_1p_3)$ denotes the distance from p_1 to p_3. The cross-ratio is well known to be invariant under perspective projection, and can be used as indexing function for object recognition. Though the cross ratio τ depends on the order of points in the argument list, there exists a rational function of τ which is independent of the order of the points. This function is the $\mathbf{j} - invariant$ [22], defined by

$$j(\tau) = \frac{(\tau^2 - \tau + 1)^3}{\tau^2(\tau - 1)^2} \tag{3}$$

Using the $\mathbf{j} - invariant$ function can simplify the process of object recognition.

Property 4. Given: (1) four pairs of corresponding model (planar object) and image points, (2) the distance between the optical center and the image plane, and (3) the image center, the 3-D location of the optical center relative to the coordinate system of the object plane can be uniquely determined [23]. The consistency between the orientation obtained by this property and that obtained by the axis and convergent point of a convergent symmetry is a constraint used in our verification process for confirming the correctness of object recognition.

B Convergent Symmetry

Given a point, P_i, on the boundary of a flat symmetric object, there exists a corresponding symmetric point, Q,, on the boundary of the object which is the reflection of P, with respect to a line, called the axis of symmetry (A). The line connecting P, and Q;, called the line of symmetry, intersects A at point M_i, the middle of segment $\overline{P_iQ_i}$, and is ortlhogonal to A. It is well known that, under perspective projection, lines remain lines, and a set of parallel lines that are not parallel to the image plane converge to a point. Therefore, under perspective projection, the lines of symmetry project to a set of lines, called lines of convergent symmetry, which converge to a point, called the convergent (vanishing) point (v). The axis of symmetry remains a line, called the axis of convergent symmetry (S), which is no longer the locus of the midpoints of the lines of convergent symmetry in the image (see figure 4). However,

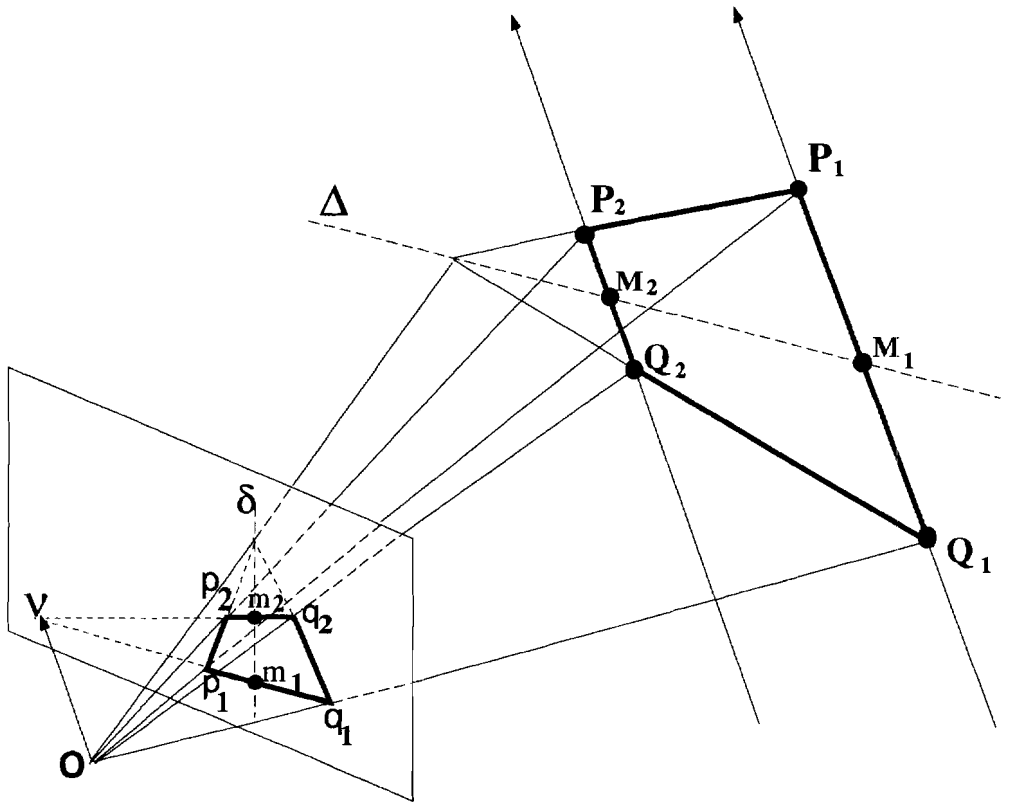


Figure 4: An example of convergent symmetry. Note that M_1 and M_2 are rnidpoints of segments $\overline{P_1Q_1}$ and $\overline{P_2Q_2}$, but m_1 and m_2 are not midpoints of segments $\overline{p_1q_1}$ and $\overline{p_2q_2}$, respectively.

some useful properties exist and allow us to find 6 and ν.

Figure 5 (a) shows a flat symmetric object, where (P_1, Q_1) and (P_2, Q_2) are two pairs of symmetric points. Figure 5 (b) shows a convergent symmetry of (a), in which (p_1, q_1) and (p_2, q_2) are the projections of (P_1, Q_1) and (P_2, Q_2), respectively.

Property 5. Given that (P_1, Q_1) and (P_2, Q_2) are two pairs of symmetric points, the intersection points, A_1 and A_2, of the two pairs of lines (P_1P_2, Q_1Q_2) and (P_1Q_2, P_2Q_1) will lie on the symmetric axis A. Since the intersection points are invariant under perspective projection, the projections of the two pairs of lines, (p_1p_2, q_1q_2) and (p_1q_2, p_2q_1), also intersect at two points, a_1 and a_2, along the axis 6 of convergent symmetry [14].

Property 6. Since lines P_1Q_1 and P_2Q_2 are parallel, their projections p_1q_1 and p_2q_2 intersect

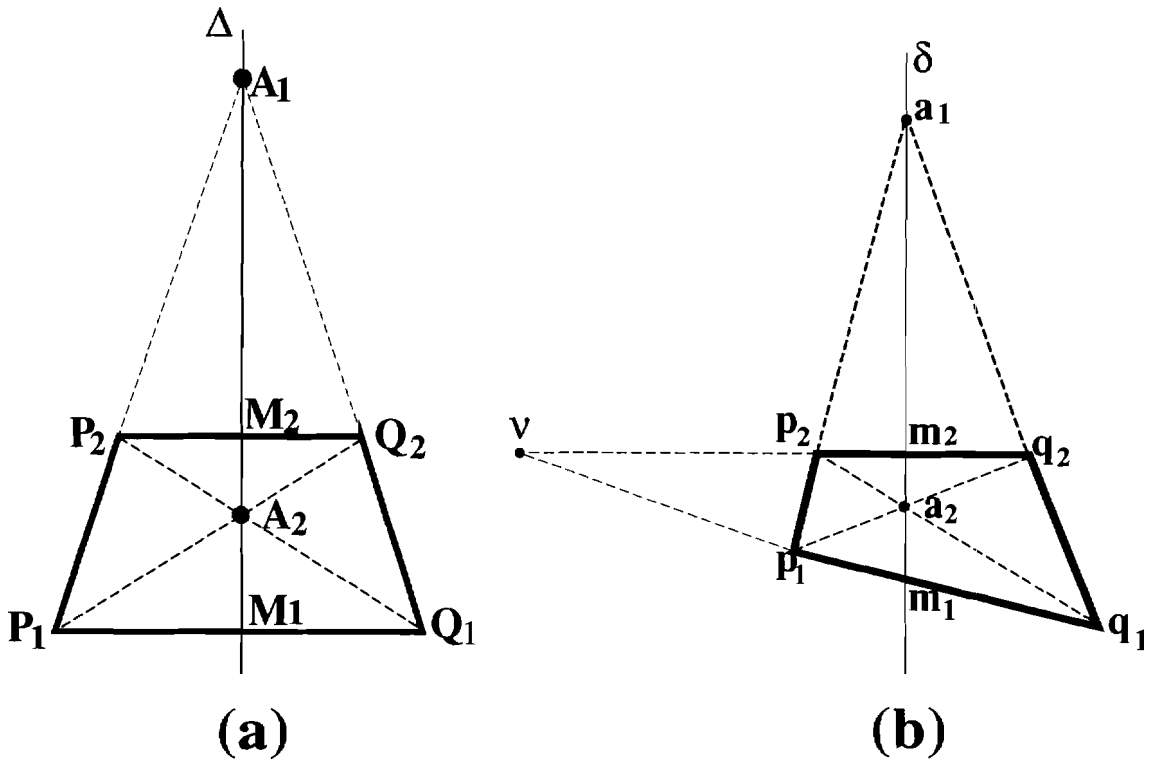


Figure 5: Determination of the axis and convergent point of a convergent symmetry from paired segments.

at the convergent point ν [14].

Definition Two image points (p, q) resulting from the perspective projection of two symmetric points (P,Q) of S will be called paired points. Furthermore, two segments $\overline{p_1p_2}$ and $\overline{q_1q_2}$ joining points of two pairs of paired points (p_1, q_1) and (p_2, q_2) will be called paired segments.

Property 7. Since the boundary of a flat symmetric object can be represented by a sequence of points, a number can be assigned to each boundary point according to its position in the sequence. For instance, if the boundary points are stored in an array, then the index of the array corresponding to each point can be used as the number. With this identification, any two pairs of symmetric distinguished points (p_i, q_i) and (p_j, q_j), represented by their associated numbers (i_1, i_2) and (j_1, j_2), must satisfy the following ordering; constraint: the numbers i_1 and i_2 should be either both inside the range from j_1 to j_2 or both outside the range (see Figure 6). This constraint is used in finding the corresponding distinguished points (paired points) of a convergent symmetry.

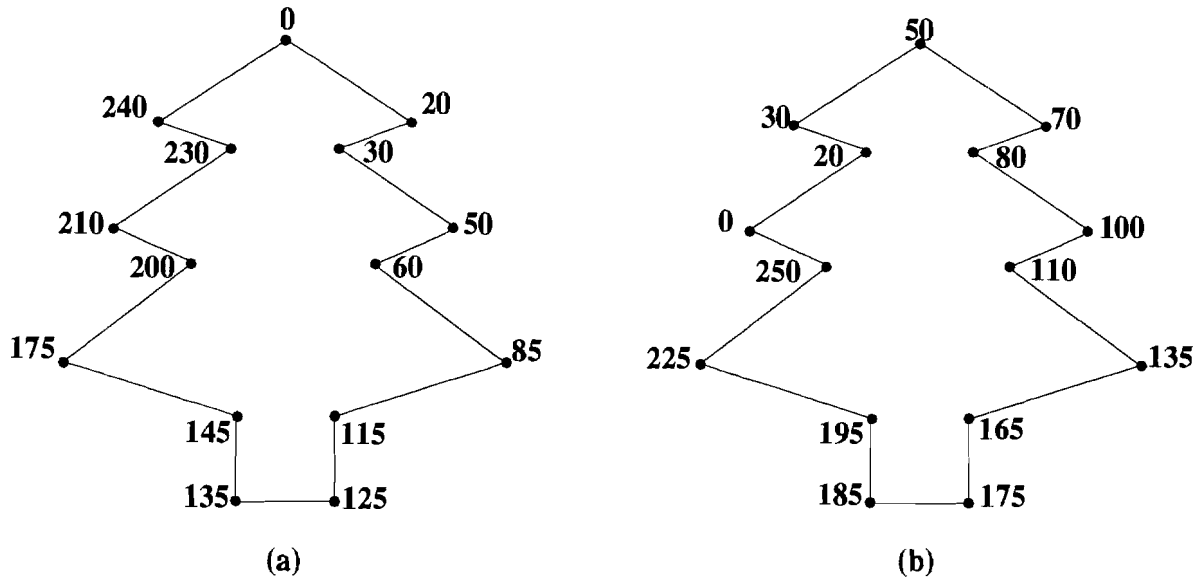


Figure 6: An ordering constraint. The numbers associated with the distinguished points (corners) of the tree patterns are roughly assigned clockwise. The difference between (a) and (b) is the position of the initial point with number 0 (the initial point is not fixed when corresponding contours are extracted from images). In both (a) and (b), all pairs of symmetric distinguished points satisfy the ordering constraint described in property 7. For instance, in (a) **(30,230)** and (85,175) are two pairs of symmetric distinguished points and they satisfy the ordering constraint because **30** and **230** are both outside the range from 85 to 175, and 85 and 175 are both inside the range from **30** to **230.**

Once the axis δ and the convergent point ν of a convergent symmetry have been found, the orientation of the underlying object can be uniquely determined.

Property 8. Let u be the direction vector of the axis A, and Π be the plane passing through the optical center O and the axis δ (note that Π also passes through A). a and b are two distinct points on δ, and $O\nu$ is the direction of the lines of symmetry. Since u is orthogornal to the lines of symmetry, u is also orthogonal to $O\nu$. As A is in the plane Π, u is orthogonal to $O\nu$ and $Oa \times Ob$. If $O\nu$ is not orthogonal to Π, then direction u can be obtained by $O\nu \times (Oa \times Ob)$ and the orientation of the object is completely determined by $u \times O\nu$; otherwise, $Ov \times (Oa \times Ob)$ is null, and the orientation of the object cannot be recovered from the axis and convergent point of the convergent symmetry [14]. Note that the focal length (i.e., the distance from optical center to image plane) has to be known in order to

obtain $O\nu$.

Based on the above properties, let us consider a flat symmetric object having some distinguished points. Since the object is symmetric, the distinguished points are paired and will be projected as paired points in the image. If we can find the distinguished points in the image and determine the paired points and paired segments, then the axis and the convergent point of the convergent symmetry can be estimated. That is, the orientation of the object can be recovered, except for some special cases described in property 8. Moreover, by using the cross ratio of the intersection points of the paired segments as an indexing function for a model database, we have developed an efficient method for recognizing flat symmetric objects.

III Finding the axis and convergent point of a convergent symmetry

In this section we present a technique for finding the axis δ and convergent point ν of a convergent symmetry. The first step involves finding the distinguished points (including bitangents, corners, and inflection points) of the image contour. Points and segments are then paired together using a Hough transformation subject to the ordering constraint described previously. Finally, a symmetric measurement based on back projecting paired points to **3-D** space is performed in order to obtain the best estimation of δ and v.

A Finding distinguished points

There are many methods available for finding distinguished points of an image contour [24, 25, 26, 27, 19]. The method in [19] uses a Hough transformation to map the tangent line at each contour point into (θ, γ) Hough space, where θ is the orientation of the line and γ is the distance from the image center to the line. It then checks the Hough space for cells containing more than one line, which are bitangents. Since this method is very efficient, we use it to find bitangents of the contour. A method based on multiscale contour approximation [24] is used to find corners and inflection points. The complexity of this method is proportional to the number of scale levels used.

Since the method of finding bitangents is more efficient than that of finding corners and

inflection points, the bitangents of an image contour are located first. If the number of bitangents found is sufficient for estimation of the axis δ and convergent point ν of convergent symmetry and for object recognition, other distinguished points do not have to be located. Though one pair of *paired segments* is necessary and sufficient to determine δ and v, we want more pairs in order to perform object recognition (described in section IV).

B Determining paired points and paired segments

After finding the distinguished points of a contour, we are ready to determine *paired points* and *paired segments*. For each pair of segments $\overline{p_i p_j}$ and $\overline{q_i q_j}$, where p;, p_j, q_i, and q_j are distinguished points, one possible axis of convergent symmetry can be computed using property 5. We then vote the computed axis onto (6, y) Hough space, where 6 is the orientation of the axis and y is the distance from the image center to the axis. Each cell in the Hough space contains two lists: listl and list2. These lists contain the corresponding pairs of segments. When the computed axis is voted to a cell, the corresponding pair of segments is checked with the pairs of segments already in listl to see whether the pair satisfies the ordering constraint (property 7). If it does, then it is pushed into listl. Otherwise it is pushed into list2. After the voting process is done, the cell with longest listl determines δ and v.

The ordering constraint effectively prevents false axes from clustering in the Hough space. However, it may happen that some pairs of segments which are not *paired segments* are pushed into the listl which is supposed to contain only *paired segments*. As a result, some *paired segments* appearing later cannot enter the list. That is why pairs of segments which do not satisfy the ordering constraint are kept in list2. The misclassified *paired segments* will be selected from list2 and added to listl during the verification step.

C Verifying the axis and convergent point of a convergent symmetry

We know that one pair of *paired segments* is sufficient for finding the axis δ and convergent point ν of a convergent symmetry. However, since there are always some errors and ambiguities resulting from segmentation and finding distinguished points, the pairs of segments in the longest listl might not be perfect *paired segments*, and should be verified in order

to select the best *paired segments*. This *paired segments* is then used to compute the axis and convergent point of the convergent symmetry by properties *5* and 6, thus recover the orientation of the observed object by property 8.

For each pair of segments $\overline{p_i p_j}$ and $\overline{q_i q_j}$ in the longest list1, we compute the corresponding δ and ν by properties *5* and 6. Then the estimated orientation of the observed object can be calculated by property 8. Knowledge of the estimated orientation of the object enables us to project those pairs of segments in the longest list1 and the computed axis δ black to **3-D** space by property 1. We then measure whether the back projected pairs of segments are symmetric with respect to the back projected axis in **3-D** space. Recall that by definition in section II, *paired segments* are the perspective projections of a pair of symmetric segments with respect to some symmetric axis in **3-D** space. Therefore, when we back project *paired segments* to 3-**D** space, the resulting segments should be symmetric with respect to the back projected axis. To see how close to symmetric a pair of corresponding segments is, we need only to compute a symmetric measure of its corresponding end points. That is the actual symmetric point of an end point with respect to the axis is calculated, the distance *(d)* between the actual symmetric point and the other corresponding end point is then computed and is considered as an error measurement. The measure of the error of symmetry of a pair of segments is then defined as:

$$e_i = \frac{1}{n} \sum_{j}^{n} d_j \tag{4}$$

where n is the number of pairs of segments in the longest list1. In fact, the more accurate the estimated orientation, the smaller the measure of the error of symmetry. Since we are only interested in this measure of symmetry, the depth of the object in property 1 can be chosen arbitrarily as long as the value is fixed for the entire verification process. Thus, a pair of segments which is closer to paired segments will have a more accurate estimated orientation of the object; as a result, its measure of error of symmetry will be smaller. Upon finishing the verification process, the pair of segments with the highest symmetry measure, i.e. the lowest measure of error of symmetry, gives the best estimation of the axis and convergent point of convergent symmetry, and the orientation of the object.

Actually, the location of the axis and convergent point can be further refined by using a

weighted average method given by:

$$\overline{m} = \frac{\sum_i^n m_i w_i}{\sum_i^n w_i} \tag{5}$$

where n is the number of pairs of segments remaining in the longest listl after the verification process; m; stands for the axis and the convergent point corresponding to the ith element in the list1; $w_i = (\frac{1}{e_i})^r$ and e_i denotes the measure of error of symmetry of the ith element of the list1. Currently, r = 2 is used. We have tested this parameter for values from 1 to 10. According to our experiments, the results obtained from this method are always better than that obtained from the pair with minimum error of symmetric measurement.

As we mentioned previously, it might happen that some pairs of segments in the longest listl are incorrectly paired. This might cause some pairs of *paired segment:?* appearing later to fail the ordering constraint and be pushed into list2 of the cell in the Hough space. This situation can be refined as follows:

- We can adjust the resolution of the Hough space so that false pairs will not be mapped to the cell in which *paired segments* are supposed to be. However, increasing the resolution of the Hough space might cause *paired segments* to be spread over several neighboring cells. Therefore, we have to check the 8-neighbors of the maximum cell in order to get all the *paired segments*.

- The false pairs of segments in the longest listl are eliminated in the verification step because the errors of their symmetric measurement are very large. After those false pairs of segments are eliminated, the pairs of segments in list2 will be added to listl if they satisfy the ordering constraint and their measure of error of symmetry are sufficiently small.

- If distinguished points of a convergent symmetry are properly located, the longest listl in the Hough space is always the correct one containing *paired segments* because of the ordering constraint. However, when noise and occlusions are present in the image, the distinguished points may not be located properly. As a result, the longest listl in the Hough space may not be the one containing correct information. In this case, we can

collect all the listl in the Hough space with length exceeding some given threshold, and perform the verification process for all the collected list1 to determine the correct list1. Since the number of distinguished points of a contour is relatively small and the verification process described above is very efficient, the time for verifying all the collected listl is reasonably small. By performing this search, we are able to detect multiple axes of convergent symmetry by sorting the verification results of all the collected listl and choosing ones with smaller errors.

IV Recognizing and locating a flat symmetric object from a single perspective image

After the verification process, the pairs of segments in the longest listl, having small errors of symmetric measurement, are likely to be the correct paired segments, and are selected for object recognition. Since they are paired segments, the two intersection points (a_1 and a_2 in Figure *5)* of each pair are likely to lie on the axis 6 of convergent symmetry by property *5;* therefore, those intersection points can be used to calculate cross ratios by property **3.** Since the cross ratios are invariant under perspective projection, if we create a model database with the cross ratios described above as an indexing function, we will be able to recognize flat symmetric objects efficiently.

A Computing the indexing function

In our approach we use two pairs of paired segments to obtain a j – invariant cross ratio because each pair gives two intersection points along 6. In this way we can recognize flat symmetric objects so long as they have at least two pairs of paired segments. Note that one pair of paired segments cannot be used to get a unique cross ratio. Though there are two other points (m_1 and m_2 in Figure *5)* on 6 that can be obtained from each paired segments (i.e., the intersection points of the lines joining two pairs of corresponding paired points and the axis δ), the cross ratios of the four points obtained this way will be the same for every pair of segments.

B Model acquisition

In our recognition system, models are acquired directly from images (they may also be acquired from CAD models). For each object, an image is taken under the situation that the object is not occluded and is parallel to the image plane. The cross ratios described above are computed and stored as keys in the indexing database. Associated with each cross ratio, only a small amount of information is required to be stored in the indexing database, such as the name of the object, the two pairs of *paired segments* which generate the cross ratio, and the height of the object (the distance between the intersection points of the axis of symmetry and object boundary). The two pairs of *paired segments* are used for hypothesis verification, while the height of the object is used for locating the object after it is recognized. Note that each *paired segments* is stored in a special order so that the correspondence between model and image *paired segments* can be easily established in the verification step. For instance, the segments are ordered according to their positions with respect to the line defined by their intersection points, while the positions of the distinguished points connected by the two segments can be ordered by the intersection point (not the one obtained by cross construction) of the two segments. The degenerate case where two segments are parallel should be handled differently.

C Hypothesis generation and verification

Given an image of an object to be recognized, the same procedure is used to compute the cross ratios of the image. For each cross ratio, we match it against those in the database. Every time it matches a model with the same cross ratio, a hypothesis is formed. Ideally, the cross ratios computed from the corresponding two pairs of *paired segments* should be the same. However, in practice there are always errors and ambiguities resulting from image formation, segmentation, and locating of distinguished points. These uncertainties propagate during the calculation of cross ratios. For instance, when paired segments are nearly parallel, a small amount of error in the positions of distinguished points will cause a big variation in the position of intersection point which in turn will affect the accuracy of the computed cross ratio. Thus, some error tolerance should be given for matching. Currently, a 5 percent

error bound is used.

As the number of models increases, the cross ratios from different models may be similar or the same. This results in some degree of ambiguity during the matching process, Although the final identification may be made by voting for the object with the greatest number of matches, objects with a small number of cross ratios (i.e., they do not have many distinguished points) may be misclassified. Therefore, some verification process should be performed in order to get correct recognition results.

In [19], it was stated that the matching process was implemented by using a hash-table technique and that the final identification was made by voting for the object with the greatest number of returns from the hash-table. We believe that the reason for small errors in the presented method is because the image contours of a rotationally symmetric surface do not have high perspective effects. In our case, the recognition results are always incorrect if we match each computed cross ratio with two nearest neighbors in the indexing database, and vote for the object with the greatest number of matches.

Our verification of generated hypotheses is based on orientation consistency checking. Recall that one pair of paired segments is sufficient for recovering the orientation. of the observed object. Moreover, the orientation of an object can also be recovered by using four pairs of corresponding model and image points (by property 4). Typically, each cross ratio is computed from two pairs of paired segments where each pair has four distinguished points. A correspondence between model and image points can be obtained by careful arrangement as described in section B. Therefore, by associating the *paired segments* obtained from the image with that stored in the indexing database, we can check whether the orientation of the object recovered by paired segments is consistent with that recovered by the four pairs of corresponding model and image points during the matching process. This constraint effectively eliminates false matches, and improves the accuracy of the final identification. These associated *paired segments* of images and models can also be used to locate objects after objects have been recognized.

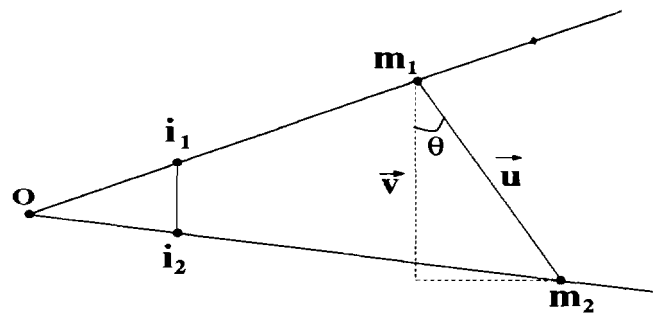


Figure 7: A method for locating the object. Let $i_1 = (x_1, y_1, -f)$ and $i_2 = (x_2, y_2, -f)$ be two intersection points of a paired segments of an image, and $m_1 = t_1(x_1, y_1, -f)$ and $m_2 = t_2(x_2, y_2, -f)$ be two intersection points of the corresponding paired segments of the corresponcling model. Let the line represented by point m_1 and direction $\vec{v}$ be the intersection of two planes: one is the plane determined by points O, i_1, and i_2; the other is $z = t_1 * (-f)$. Since the length and the orientation of $\vec{u} = m_2 - m_1$ are known, we have the following two equations: $cos\theta = \frac{\vec{u}\cdot\vec{v}}{|\vec{u}||\vec{v}|}$, $|m_2 - m_1| = |\vec{u}|$. The above equations can be solved for t_1 and t_2, thus the location of the object can be completely determined.

D Locating the flat symmetric object

As we have mentioned, the orientation of an observed flat symmetric object can be determined from the axis and convergent point of its convergent symmetry by property 8. After the object has been recognized, the information available from the model database enables us to locate the object. For instance, the distances of two intersection points of the corresponding image and model paired segments allow us to recover the depth information(see figure 7). Thus the location of the object is completely determined. According to our experiment, the location obtained this way is more accurate than that obtained by using property 4.

V Experimental results

The algorithm was implemented on a DEC 5000/200 workstation using the Common LISP programming language. Some flat symmetric objects (some of them are similar to those used in [14]) were drawn by hand. The images of these objects were taken from different viewpoints using a SONY XC-711 camera. A Canny edge operator and curve tracking program were used to extract contours from these images. Note that the input of our algorithm does not have to be a closed contour. A sequence of edges (with some small gaps in between) which make up a convergent symmetric contour will do.

Figure 8 presents an example showing that the axis of a convergent symmetry may not be correctly found without using the ordering constraint. Figure 9 shows that our technique is able to detect multiple axes of a convergent symmetry. Some images of a curved object taken from different viewpoints were processed. The results of the orientation recovery are shown in Table 1. From this table, one can see that the error is within 5 degrees. We are currently improving the calibration process of our experimental environment in order to increase the accuracy of our results.

The figures of four objects are shown next. Figures 10-(a), 11-(a), 12-(a), and 13-(a) are the images of the curve, club, butterfly, and leaf objects. Figures 10-(b), 11-(b), 12-(b), and 13-(b) show the extracted contours from the images in Figures (a) and the correctly detected axes. The darker contours in figures 10-(c), 11-(c), 12-(c), and 13-(c) are (obtainedby back projecting (based on the result of recognition and localization processes) t'he corresponding image contours to 3-D space, rotating the normals of the objects so that their normals coincide with the z-axis, and orthographically projecting the resulting objects to the image plane. The dashed contours in the figures are the corresponding model contours from the model database. One can see that the reconstructed contours have almost the same shapes as those of the original objects. Figures 15 and 14 show that our approach of finding δ and ν is robust to noise and partial occlusion as long as some pairs of corresponding distinguished points can still be located.

Tables 2 and 3 show typical results of recognition using only the four objects just shown. The difference between the two tables is that Table 2 is obtained when the inflection points

and corners of the leaf object are included in the process while Table 3 is obtained when those are excluded. The information in the two tables show that voting for the object with the greatest number of matches is not always correct, even for a database containing only four objects. However, the constraint checking of whether the orientation of object recovered from δ and ν is consistent with that recovered by four pairs of corresponding model and image points, effectively eliminates those false matches, and the correct identification is always made. There are still some false matches after the orientation consistency checking. However, these false matches do not affect the final identifications, and can be completely eliminated by comparing the reconstructed contours and the corresponding identified model contours as shown in figures 10-(c), 11-(c), 12-(c), and 13-(c).

Figure 16-(a) is an image of a box which has some letters on its surfaces. Figure 16-(b) is the result after a Canny edge operator is applied on the image. The box was intentionally overexposed so that the three intersection lines between three visible surfaces are not visible in the image. However, our technique is able to extract the symmetric letters on the surfaces and estimate the locations of the corresponding surfaces (see figure 17). From the recovered orientations of the three visible surfaces, we calculate three angles between surface normals. The results are 87.59°, 89.33°, and 89.27" which are very similar to the actual angles of the box (all 90°).

VI Further Enhancements

As mentioned in section IV, we use two pairs of *paired segments* to compute a $j-invariant$ cross ratio as an index for an object. However, when the number of *paired segments* increases, the number of the indices of an object will increase exponentially. As can be seen from Table 2, the number of the indices of the leaf object is much larger than that of other objects. This is because the number of distinguished points and *paired segments* of the leaf object is large. Matching complexity can be greatly reduced if the cross ratio can be computed from one *paired segments* instead of two. In the next paragraph, we describe a method for accomplishing this task.

Suppose that we have already found the axis of convergent symmetry. There are always two

intersection points between the axis and the convergent symmetric contour. If there are more than two intersection points, we can choose the two points which are furthest apart. These two points together with the two intersection points computed from each *paired segments* enable us to obtain a cross ratio. Besides reducing the complexity of the object recognition process, this method also increases the class of objects which can be recognized by the recognition system. This increase is due to the fact that the number of *paired segments* required for an object to be included in the model database has been reduced to one. The disadvantage of this method is that if an object is occluded so that the intersection points of the axis and the contour cannot be correctly located, incorrect recognition may occur.

VII Conclusion

We have presented a technique for finding the axis 6 and convergent point ν of a convergent symmetry and for recognizing and locating flat symmetric objects. The ordering constraint and symmetric measurement make the process of finding 6 and ν efficient and accurate. Furthermore, the orientation consistency checking improves the results of object recognition. After objects have been recognized, the information of the corresponding objects allows us to recover the depth information which is missing when the images of the objects were taken. Therefore, the localization problem can be solved. Our technique of recognizing flat symmetric objects can also be used in **3-D** object recognition, since many **3-D** objects contain planar shapes which may themselves be symmetric or contain symmetric figures. We are now investigating methods for recognizing and locating more general **3-D** objects.

References

[l.] T. Kanade, "Recovery of the three-dimensional shape of an object from a single view," Artificial Intelligence, vol. 17, pp. 409–460, 1981.

[2] S. A. Friedberg, "Finding axes of skewed symmetry," Computer Vision, Graphics, and Image processing, vol. 34, no. 2, pp. 191–200, 1986.

[3] W. G. Oh, M. Asada, and S. Tsuji, "Finding of partly occluded skewed symmetry," Proceedings 5th Scandinavian Conference on Image Analysis, June 1987, International Association for Pattern Recognition, pp. 191–200.

[4] J. Ponce, "On characterizing ribbons and finding skewed symmetries," Computer *Vision,* Graphics, and Image processing, vol. 52, no. 3, pp. 328–340, 1990.

[5] A. D. Gross and T. E. Boult, "Syman: a symmetry analyzer," *Proceedings* IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, 1991, pp. 744–746.

[6] M. Brady and A. Yuille, "An extremum principle for shape from contour," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 6, no. 3, pp. 288–301, 1984.

[7] F. Ulupinar and R. Nevatia, "Using symmetries for analysis of shape from contour," Proceedings 2nd International Conference on Computer Vision, 1988, pp. 414–426.

[8] S. Y. K. Yuen, "Shape from contour using symmetries," *Proceedings* First European *Conference* on Computer Vision, 1990, pp. 437–450.

[9] F. Ulupinar and R. Nevatia, "Constraints for interpretation of line drawings under perspective projection," Computer Vision, Graphics, and Image *processing:* Image Understanding, vol. 53, no. 1, pp. 88–96, 1991.

[10] R. T. Collins and R. S. Weiss, "Vanishing point calculation as a statistical inference on the unit sphere," Topical meeting on Image Understanding and Machine *Vision,* Optical Society of America, 1989 Technical Digest Series, 1990, pp. 400–403.

[11] A. Tai, J. Kittler, M. Petrou, and T. Windeatt, "Vanishing point detection," Image and Vision Computing, vol. 11, no. 4, pp. 240–245, 1993.

[12] M. Straforini, C. Coelho, and M. Campani, "Extraction of vanishing points from images of indoor and outdoor scenes," Image and Vision Computing, vol. 11, no. 2, pp. 91–99, 1993.

[13] E. Lutton, H. Maitre, and J. Lopez-Krahe, "Contribution to the determination of vanishing points using hough transform," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 4, pp. 430–438, 1994.

[14] R. Glachet, J. T. Lapreste, and M. Dhome, "Locating and modelling a flat symmetric object from a single perspective image," Computer Vision, Graphics, and Image processing: Image Understanding, vol. 57, no. 2, pp. 219–226, 1993.

[15] M. Dhome, R. Glachet, and J. T. Lapreste, "Recovering the scaling function of s shgc from a single perspective view," IEEE Computer Society Conference on *Computer* Vision *and* Pattern Recognition, 1992, pp. 36–41.

[16] J. Ponce, D. M. Chelberg, and W. B. Mann, "Invariant properties of straight homogeneous generalized cylinders and their contours," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 9, pp. 951–966, 1989.

[17] M. Richetin, M. Dhome, J. T. Lapreste, and G.Rives, "Inverse perspective transform using zero-curvature contour points: Application to the localization of some generalized cylinders from a single view," IEEE Transactions on Pattern *Analysis* and Machine Intelligence, vol. 13, no. 2, pp. 185–192, 1991.

[18] T. Nakamura, M. Asada, and Y. Shirai, "A qualitative approach to quantitative recovery of shgc's shape and pose from shading and contour," IEEE Computer *Society* Conference on Computer Vision and Pattern Recognition, 1993, pp. 116–122.

[19] J. Liu, J. Mundy, D. Forsyth, A. Zisserman, and C. Rothwell, "Efficient recognition of rotationally symmetric surfaces and straight homogeneous generalized cylinders," IEEE Computer Society Conference on Computer Vision and Pattern *Recognition*, 1993, pp. 123–128.

[20] S. **A.** Shafer, T. Kanade, and J. Kender, "Gradient space under orthography and perspective," Computer Vision, Graphics, and Image processing, vol. 24, pp. 182–199, 1983.

[21] J. L. Mundy and A. Zisserman, Geometric Invariance in Computer Vision. MIT Press, 1992.

[22] R. Hartshorne, Algebraic Geometry. Springer-Verlag, 1977.

[23] M. **A.** Fischler and R. C. Bolles, "Random sample consensus: **A** paradigm for model fitting with applications to image analysis and automated cartography," Communications of the ACM, vol. 24, pp. 381–395, 1981.

[24] A. Bengtsson and J.-O. Eklundh, "Shape representation by multiscale contour approximation," IEEE Transactions on Pattern Analysis and Machine *Intelligence*, vol. 13, no. 11, pp. 85–93, 1991.

[25] K. Shon, W. E. Alexander, J. H. Kim, Y. Kim, and W. E. Snyder, "Curvature estimation and unique corner point detection for boundary representation," *IEEE* International Conference on Robotics and Automation, 1992, pp. 1590–1595.

[26] K.-B. Eom and J. Park, "Contour models for curvature estimation and shape decomposition," IEEE *11th* International Conference on Pattern Recognition, 1992, pp. 393–396.

[27] R. Legault and C. Y. Suen, "A comparison of methods of extracting curvature features," IEEE *11th* International Conference on Pattern Recognition, 1992, pp. 134–138.

Rotation about x and y axes	Recovered angles (1)	Recovered angles (2)
x-15 y-15	x-11.5 y-20.2	x-12.7 y-17.5
x-30 y-30	x-28.8 y-32.8	x-30.1 y-31.1
x-45 y-45	x-44.8 y-41.7	x-44.5 y-43.3
x-30 y-60	x-33.3 y-54.9	x-30.0 y-59.8
x-60 y-60	x-61.9 y-56.0	x-62.3 y-56.1

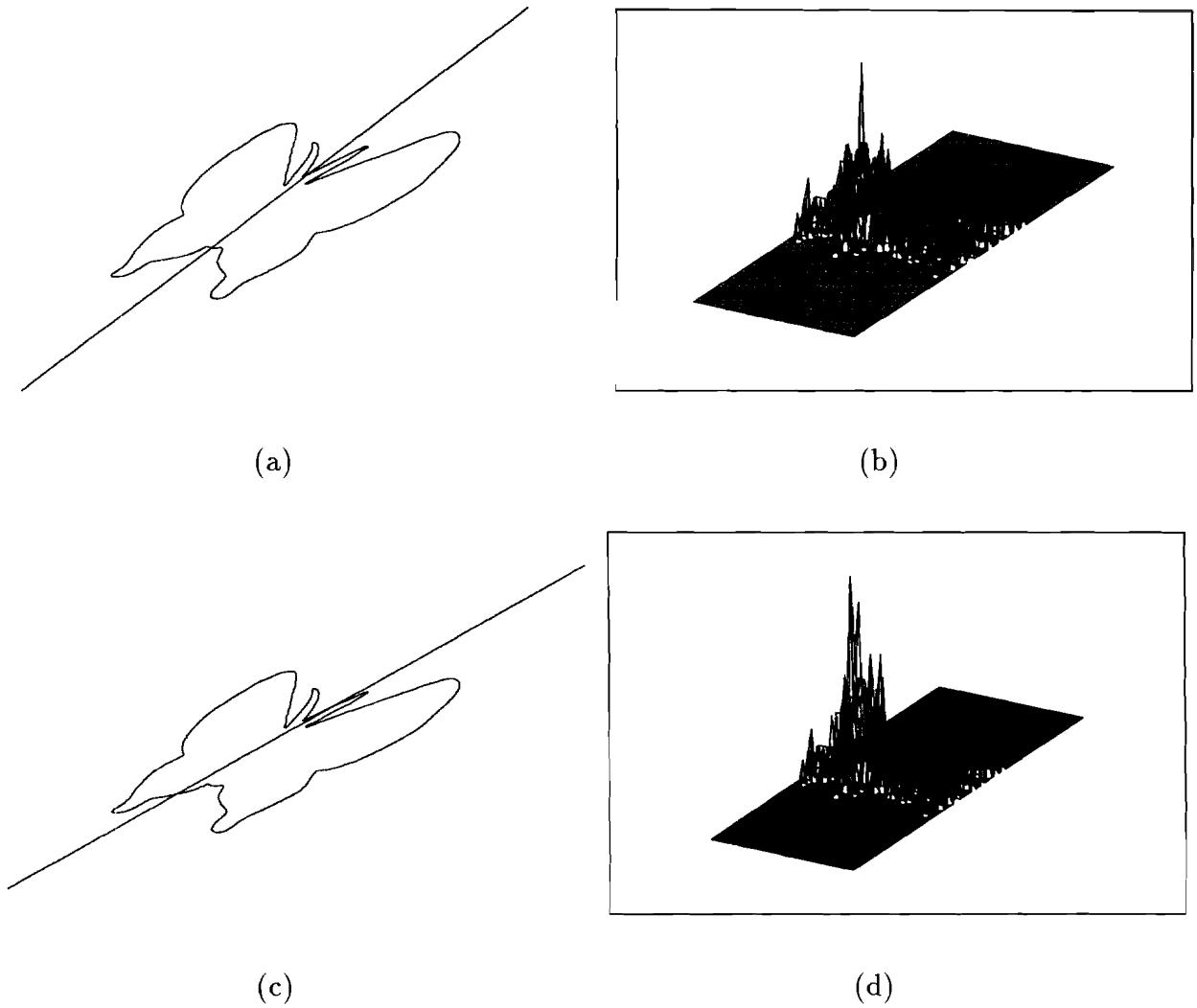
Table 1: Orientation recovered from curved images taken from different viewpoints. Some images of the curved object are taken from different viewpoints. Actually, we fix the camera and rotate the object about the x and y axes of the camera coordinate. The first column contains the rotated angles (in degree) about x and y axes when we took the images. The second column contains the recovered angles computed from the pair of segments with minimum error of symmetric measurement, whereas the third column contains the recovered angles by using the weighted average method.

Image contour	Matched list	Filtered list	Result
curve (101)	(101 9) (102 11) (103 25) (104 54)	(101 3)	curve
club (102)	(101 38) (102 70) (103 168) (104 286)	(102 4) (103 2)	club
butterfly (103)	(101 29) (102 24) (103 72) (104 167)	(103 3)	butterfly
leaf (104)	(101 279) (102 331) (103 1049) (104 2115)	(104 12)	leaf

Table 2: The results of recognition when the inflection points and corners of the leaf object are included in the process. The first column contains the names and identi~ficatiomumbers of the objects corresponding to image contours to be recognized. The cross ratios generated from those image contours were matched with the pre-created cross-ratio database. The resulting matched lists are contained in the second column. The elements in the list are the identification number of an object and the number of matches corresponding to the object. The third column contains the lists after matches which do not satisfy the orientation consistency checking are eliminated from the matched lists. The last column shows the results of recognition.

Image contour	Matched list	Filtered list	Result
curve (101)	(101 9) (102 11) (103 25) (104 11)	(101 4)	curve
club (102)	(101 38) (102 70) (103 168) (104 10)	(102 4) (1032)	club
butterfly (103)	(101 29) (102 24) (103 72) (104 20)	(103 3)	butterfly
leaf (104)	(101 51) (102 57) (103 128) (104 37)	(104 1)	leaf

Table 3: The results of recognition when the inflection points and corners of the leaf object are excluded in the process. See Table 2 for more information.



(a) (b)

(c) (d)

Figure 8: An example showing the results of the detection of the axis of a convergent symmetry. (a) and (b) The detected axis and the histogram of the corresponding Hough space when the ordering constraint is applied. The highest peak in the histogram corresponds to the detected axis. (c) and (d) The results without using the ordering constraint.

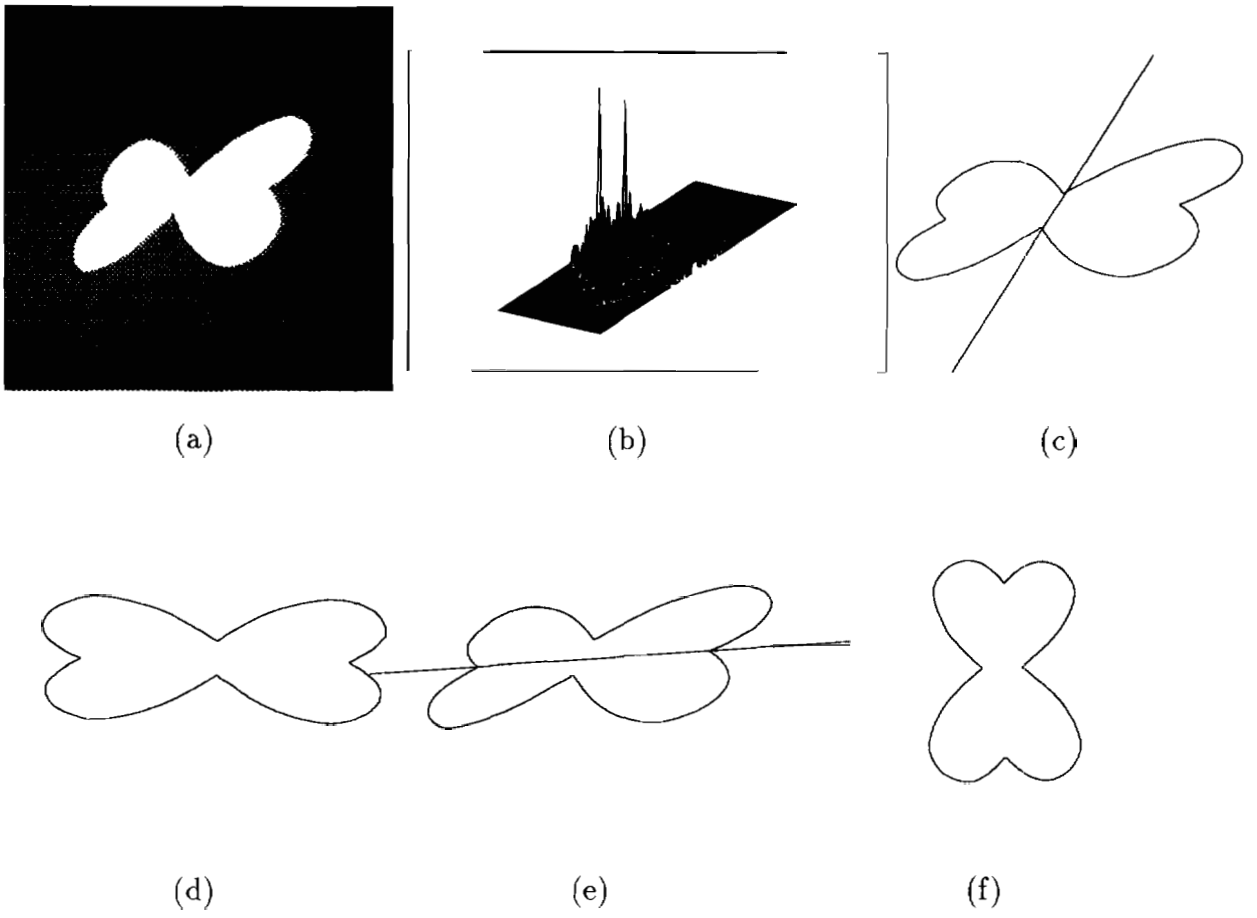
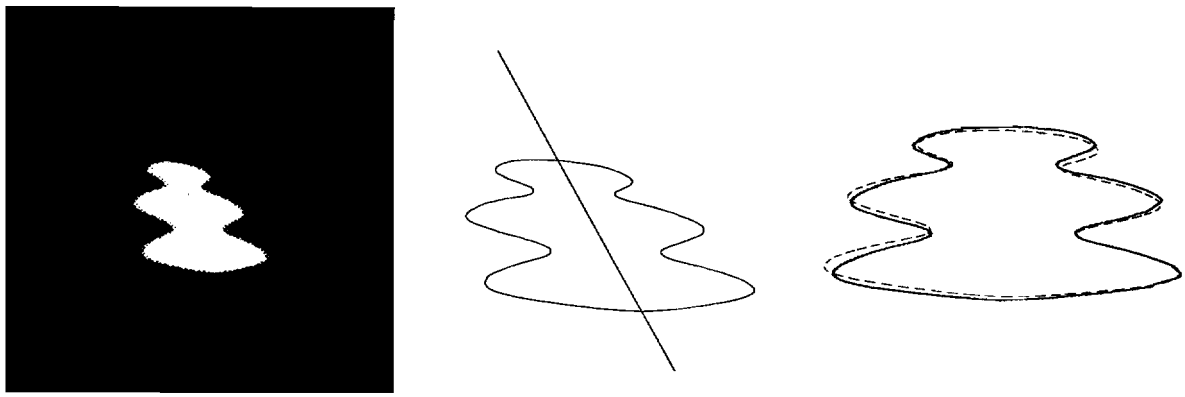
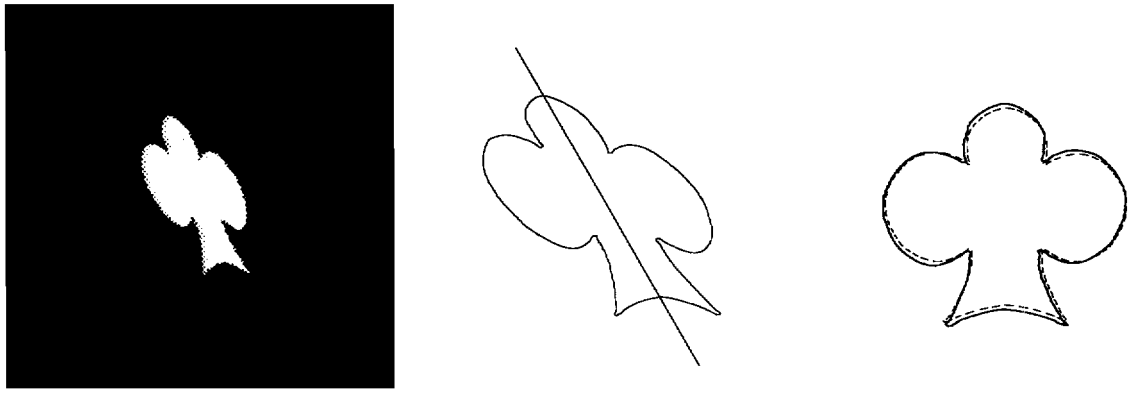


Figure 9: An example showing that multiple axes of convergent symmetry can be found. (a) The image of an object. (b) The histogram of Hough space. Two peaks in the histogram correspond to the two axes of the convergent symmetry which are shown in (c) and (e). (d) and (f) show the reconstructed contours based on the recovered orientations from the corresponding axis and convergent point.



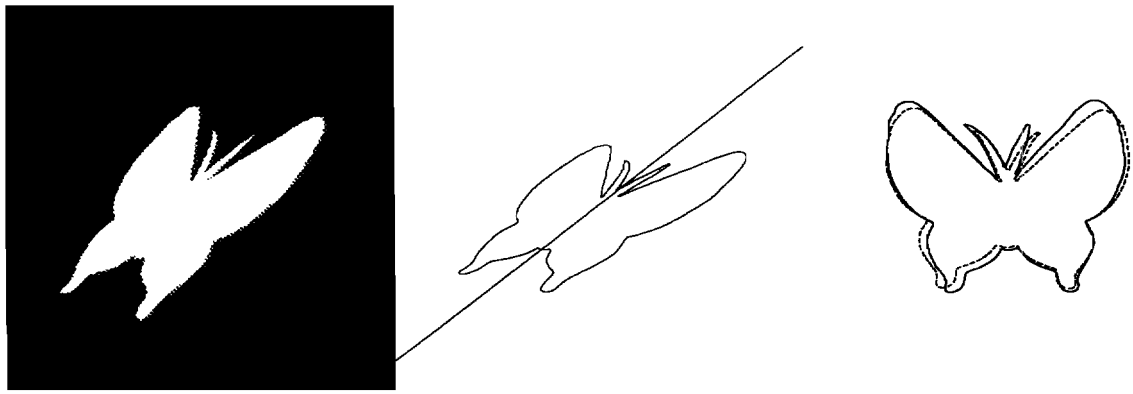
(a) (b) (c)

Figure 10: A curve image.



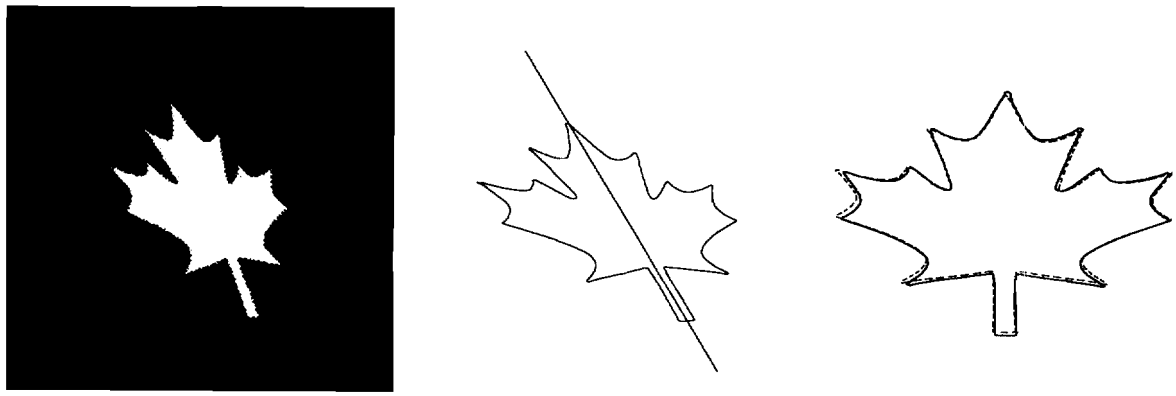
(a) (b) (c)

Figure 11: A club image.



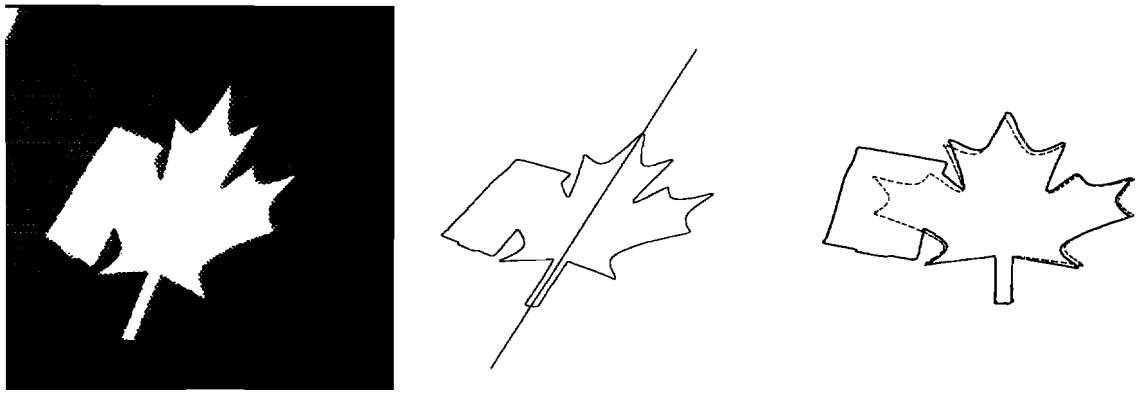
(a) (b) (c)

Figure 12: A butterfly image.



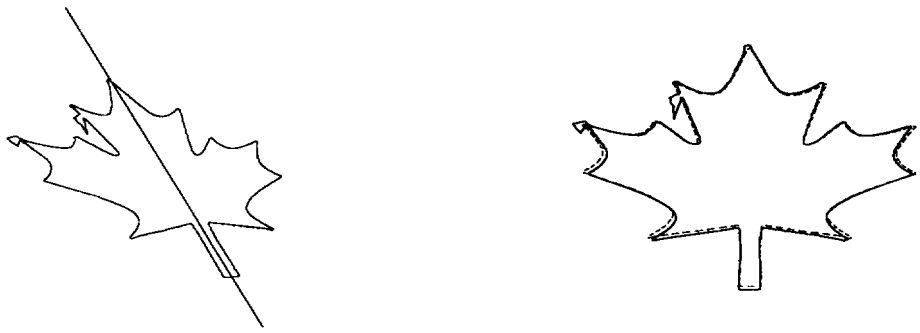
(a) (b) (c)

Figure 13: A leaf image.



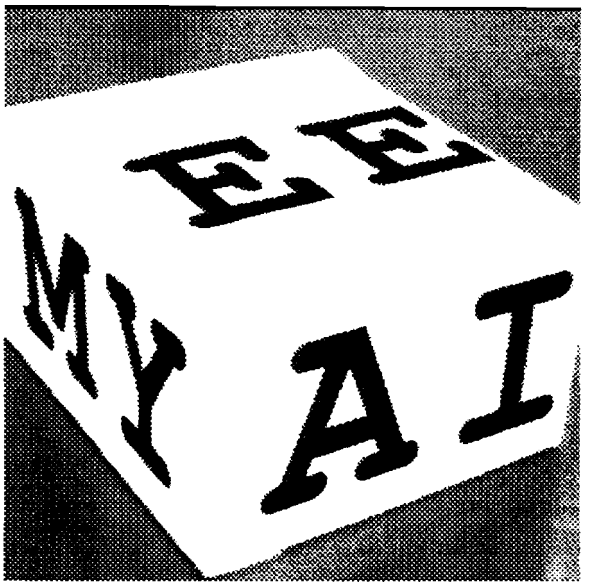
(a) (b) (c)

Figure 14: A partly occluded leaf image.

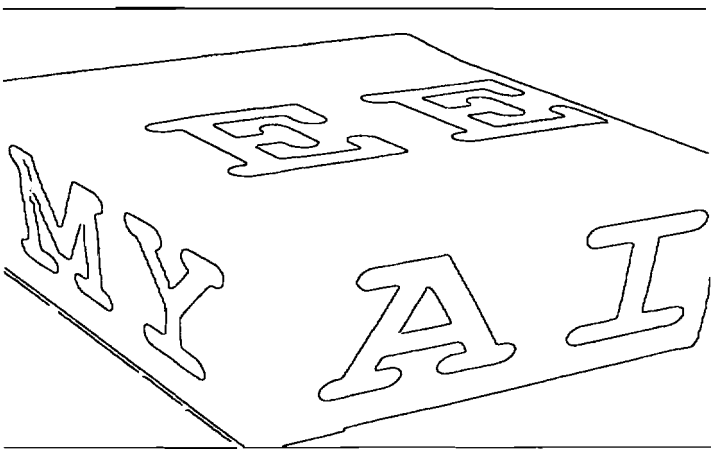


(a) (b)

Figure 15: A leaf image with noise.



(a)



(b)

Figure 16: An image of a box with some letters on the surfaces.

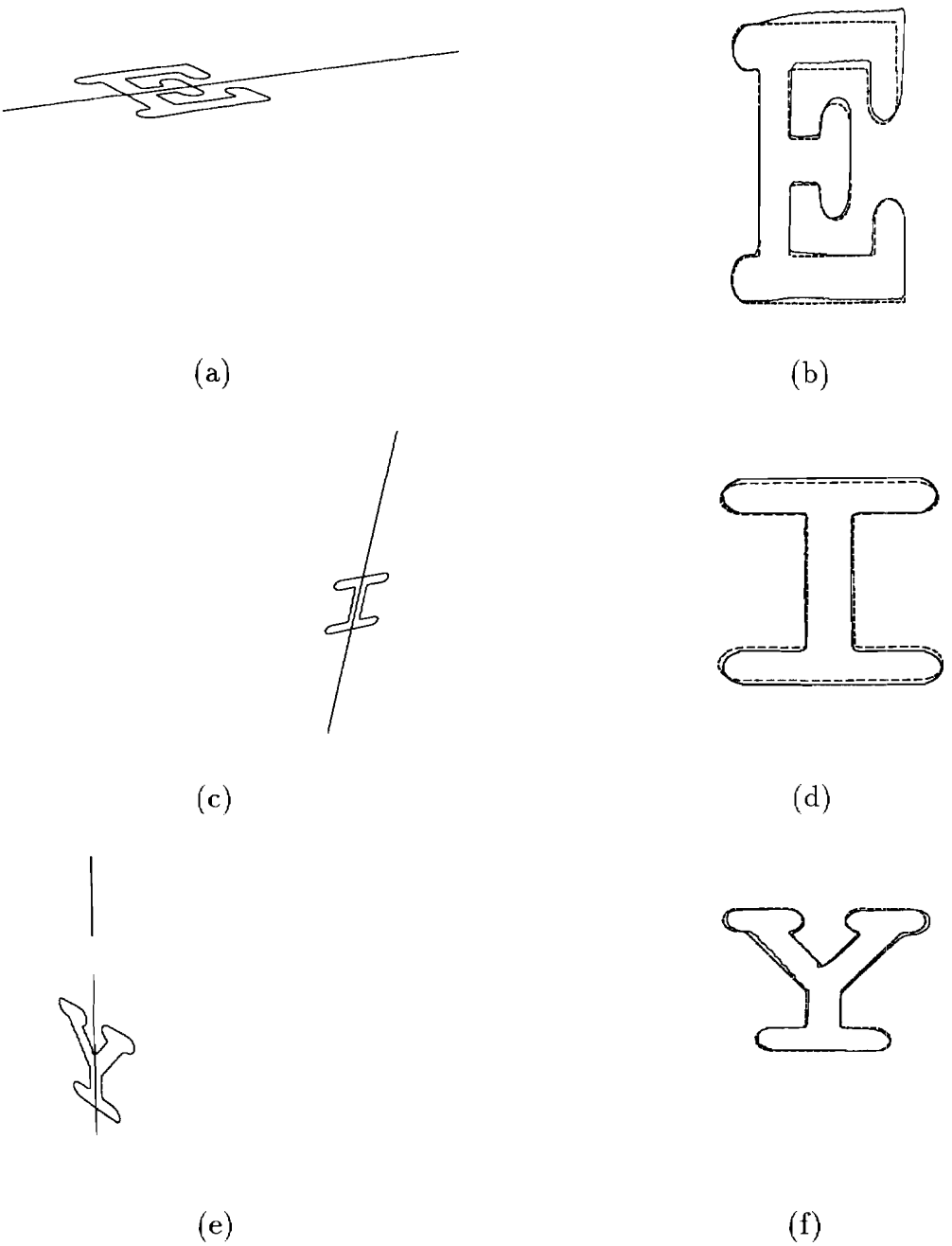


Figure 17: Letters with their axes of convergent symmetries and back projections.