コンピュータビジョン 96－2
（1995．9．28）

確率計算による平面剛体運動の解析

イリス フェルミン, 井宮 淳, 市川 熹

千葉大学 工学部 情報工学科
〒263 千葉市稲毛区弥生町 1-33
firis@icsd4.tj.chiba-u.ac.jp, imiya@ics.tj.chiba-u.ac.jp

あらまし 従来, 物体表面の基準点の動きを追跡したり, 基準点の画像上での対応が決定されていることを仮定して物体の運動変数を推定してきた. しかし, 実際の運動解析では, 必ずしも物体の基準点が事前に決定されているとは限らない. そこで, 本論文では, 対応点の事前決定を必要としない平面剛体運動の運動変数の計算法を提案する. 本論文では, 運動する平面図形の中の合同な三角形を確率的に探索して運動変数を推定する算法を提案し, その性能を評価する. 提案する手法は, ハフ変換の考えを運動解析に応用したものである. また, 提案する算法は平面図形の確率的な整合判定算法にもなっている.

キーワード 剛体運動, 運動変数, 確率算法, 対応点, 平面図形

Randomized Method for Planar Motion Detection

Iris Fermin, Atsushi Imiya*, and Akira Ichikawa
Dept. of Information and Computer Sciences, Chiba University
1-33 Yayoi-cho, Inage-ku 263, Chiba, Japan
firis@icsd4.tj.chiba-u.ac.jp, imiya@ics.tj.chiba-u.ac.jp

Abstract In this paper, we propose a randomized algorithm to estimate the planar motion parameters of a finite closed set. By randomly searching points on two shapes measured at different times, we determine the centroid and translation of the planar shapes. Taking these centroids as references, the algorithm proceeds to determine the point-to-point correspondences by randomly searching three points on each shape that form congruent triangles. After the point correspondences have been determined, the rotation is estimated by solving the rigid motion equation.

Key words rigid motion, motion parameters, randomized algorithm, point correspondences, planar shape

* To whom all correspondence should be addressed.

1 Introduction

Estimation of motion parameters is a classical problem of computer vision which has been studied by many researchers. There are two main approaches to examining time-varying images: the feature-based approach (feature correspondence) and the differential-based approach (optical flow). Both approaches can use either monocular or binocular image sequences.

Methods based on image feature positions must to deal with the problem of point correspondences. Many of these methods assume that the correspondence problem has been solved. However, the advantage of these kinds of linear algorithms is that they are fast and the uniqueness of solution is guaranteed except in degenerate cases. Methods based on optical flow must to solve nonlinear equations through iterative methods that may diverge or converge to local minima. Hence the search in the space of motion parameters is computationally expensive, although the optical flow under many conditions can be linearly approximated [1]. Likewise, other workers have tried to avoid the matching problem and the computation of optical flow by using a direct motion approach [2].

There are a number of reasons why the problem of point correspondences is difficult. For example, more than one measurement may match a predicted feature or a single measurement may match more than one feature. Also if there are occluded parts, some features can appear in one image frame and then disappear in the next frame. These ambiguities are the essence of motion correspondence problem. However, the point correspondences problem is assumed *a priori* to be solved for the majority of motion detection algorithms.

In computer vision, recognition of a planar shape by matching image features which are invariant under transformations of the image caused by motion is a well-studied problem [3], and has numerous applications. Here, we are interested in detecting the motion without knowing *a priori* the point correspondences. The approach used in this work is to take two images acquired at different times by a camera moving in a static environment and then use random search and shooting circles to a develop non-model-based motion detection algorithm. We propose an algorithm based on the randomized Hough transform (RHT) and on the rigid body motion constraint [4]. The rigid body motion constraint provides a condition that makes the problem well-posed. Furthermore, it is well known that the randomized Hough transform approach decreases the time consumed dramatically.

The present algorithm randomly searches points on each shape to estimate the centroids of the planar shapes, and then determines the translation using the centroid constraint. To solve the point-to-point correspondence problem the algorithm randomly searches three points subject to the constraint that points in each shape form congruent triangles. After possible 3-point correspondences are identified, the algorithm proceeds to estimate the rotation using the rigid motion constraint and 'voting' in the parameter space. This process continues until a limit of iteration cycles or a peak is detected in the parameter space.

The aim of this work is to develop an algorithm to estimate the motion parameters in the two-dimensional Euclidean space, which deals firstly with the problem of point correspondences, and takes advantage of the linear constraint and the higher speed provided by the randomized approach.

2 Rigid Object Motion

The term rigid body means an assembly of particles with fixed interparticle distances. Thus, in the kinematics of solid objects, the motion of an object is rigid if and only if the distance between points of the body is invariant with time. Rigid motion [5, 6, 7] can be described as the sum of translation plus rotation about an axis that is fixed in direction for short periods of time. Let R^2 be the two-dimensional Euclidean plane, and denoting by x, y be the orthogonal coordinates on R^2, we express a vector on R^2 as

$$\boldsymbol{x} = (x, y)^T, \tag{2.1}$$

and the norm of the vector as

$$|\boldsymbol{x}| = \sqrt{\boldsymbol{x}^T \boldsymbol{x}}, \tag{2.2}$$

where $\boldsymbol{x}^T\boldsymbol{x}$ is the inner product of vectors. Furthermore, $\boldsymbol{x} \in R^2$ is expressed using homogeneous coordinates as

$$\boldsymbol{\xi} = (x, y, 1)^T. \tag{2.3}$$

We call a finite closed set $\boldsymbol{V}$ on R^2 a planar shape. The rigid motion for a point $\boldsymbol{x}$ on a planar shape $\boldsymbol{V}$ is defined by

$$\boldsymbol{x}' = \boldsymbol{R}\boldsymbol{x} + \boldsymbol{a} \tag{2.4}$$

where $\boldsymbol{a} \in R^2$ is the translation vector, $\boldsymbol{R}$ is the rotation matrix such that

$$\boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}, \quad \text{and } |\boldsymbol{R}| = 1 \tag{2.5}$$

and the two-dimensional rotation matrix is defined as

$$\boldsymbol{R} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}. \tag{2.6}$$

We define $\boldsymbol{V}'$, that is the result of applying the rigid transformation to $\boldsymbol{V}$ as

$$\boldsymbol{V}' = \{\boldsymbol{x}' | \boldsymbol{x}' = \boldsymbol{R}\boldsymbol{x} + \boldsymbol{a}\}. \tag{2.7}$$

We compute the motion parameters from two shapes V and V' measured at times t_1 and t_2, respectively, and if we know three sets of corresponding points we can solve eq.(2.4). Otherwise, the rigid motion constraint is a linear equation; therefore we can apply the concept of the randomized Hough transform [4]. The Euclidean motion may be decomposed into rotation and translation; accordingly, the algorithm is composed of two main stages: first, computation of the centroid and translation and second, computation of the rotation parameters. Furthermore, we assume the rotation to be about the centroid of the planar shape, then the translation of the centroid corresponds to the translation of the planar shape.

Setting dx and dy to define an infinitesimal lenghts on x and y axes respectively, the centroids of V and V' are obtained as

$$g = \frac{\iint_V x dx dy}{\iint_V dx dy} \quad g' = \frac{\iint_{V'} x' dx dy}{\iint_{V'} dx dy}, \tag{2.8}$$

respectively. Since we have

$$\iint_{V'} dx dy = \iint_V dx dy \tag{2.9}$$

and

$$\begin{aligned} \iint_{V'} x' dx dy &= \iint_V (Rx + a) dx dy \\ &= \iint_V Rx dx dy + a \\ &= \iint_V x dx dy + a, \end{aligned} \tag{2.10}$$

we obtain the following relation.

Proposition 1 *Since*

$$g' = g + a, \tag{2.11}$$

the relation

$$a = g' - g \tag{2.12}$$

holds.

In addition, in rigid object motion the distance between two object points is conserved and three correspondence points provide enough constraints to recover the rotation parameter [8]. We randomly select three noncollinear points from each frame. Hence if these two sets of three points form congruent triangles in shapes V and V', then these two sets define point correspondences (see figure 1).

Let V be a planar shape; that is, V is a finite closed set on R^2. Denoting by ∂V the boundary of V, for a point $g \in V$, we define a set

$$K(r) = \partial V \cap \{x \mid |x - g| = r, \quad r > 0\}. \tag{2.13}$$

Furthermore, letting $|K(r)|$ be the number of elements of $K(r)$, we obtain the following theorem.

Theorem 1 *For any triplet of vectors x, y and $z \in K(r)$ we can show that $|x - y|$, $(x - z)^T(y - z)$, and $|K(r)|$ are invariant under Euclidean motion.*
(Proof.) For $x \in V$ let

$$V' = \{x' \mid x' = Rx + a\}, \tag{2.14}$$

thus

$$|x' - y'| = |x - y|, \quad \forall\, y' \in V' \text{ and } \forall\, x \in V \tag{2.15}$$

and

$$\begin{aligned} (x' - z')^T(y' - z') &= (x - z)^T(y - z), \\ &\forall\, y' \in V' \text{ and } \forall\, x \in V. \end{aligned} \tag{2.16}$$

Furthermore, $K(r) \subset V$, and by setting

$$g' = g + a, \tag{2.17}$$

we obtain

$$K'(r) = \partial V' \cap \{x' \mid |x' - g'| = r\}. \tag{2.18}$$

This leads to the conclusion that $|K'(r)| = |K(r)|$. (Q.E.D)

Theorem 1 implies that in a continuous space, we can solve the point correspondences problem by using $|K(r)|$ and $|K'(r)|$, since in most cases $|K(r)|$ and $|K'(r)|$ are finite. The set of three points is selected from the intersection of the boundary and a circle whose center is the centroid g with radius r.

After point correspondences have been established, the rotation is computed using the rigid body motion constraint. Random point selection is continued until a peak is detected in the parameter space or a limit of iteration is reached.

3 Randomized Algorithm

Primarily, for computation of motion parameters we require a digital image. Hence, we proceed to detect the image boundary, in this case using binary dilation [9], after which we can apply the algorithm.

3.1 Estimation of Centroid and Translation

The centroid $g=(\bar{x}, \bar{y})^T$ of a region V is defined as

$$\bar{x} = \frac{\int_V x dx}{\int_V dx dy} \quad \bar{y} = \frac{\int_V y dy}{\int_V dx dy}. \tag{3.1}$$

A region V can be decomposed into nonoverlapping parts, that is,

$$V = V_i \cap V_j, \quad i \neq j \text{ for } i,j = 1,2,3,...,n. \tag{3.2}$$

For $V_i \subset V$ and $V_j \subset V$,

$$intV_i \cap intV_j = \emptyset \tag{3.3}$$

where $intV$ is the interior of the region. For a region,

$$V = \partial V \cup intV. \tag{3.4}$$

Denoting by g and g_i the centroids of V and V_i, we have the relations

$$g = \frac{1}{|V|}\sum_{i=1}^{n}|V_i|g_i \tag{3.5}$$

and

$$g = \frac{|V \backslash V_n|}{|V|}\bar{g}_{n-1} + \frac{|V_n|}{|V|}g_n, \tag{3.6}$$

where $|V|$ is the area measure of the set V and

$$\bar{g}_{n-1} = \frac{1}{|V \backslash V_n|}\sum_{i=1}^{n-1}|V_i|g_i, \tag{3.7}$$

where $\backslash$ is the usual set subtraction operation. Thus, for a digital image, eqs.(3.1) and (3.6) can be written as

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n}x_i, \quad \forall x_i \in V \quad \bar{y} = \frac{1}{n}\sum_{i=1}^{n}y_i, \quad \forall y_i \in V. \tag{3.8}$$

For our random approach we rewrite these equations as:

$$g_{n+1} = \frac{n}{n+1}g_n + \frac{1}{n+1}x$$

$$g'_{n+1} = \frac{n}{n+1}g'_n + \frac{1}{n+1}x' \tag{3.9}$$

where $g_0 = 0$, $g'_0 = 0$. The algorithm randomly selects points $x \in V$ and $x' \in V'$, where V and V' are the sets of the planar shape points, then the centroid is computed until a limit of iteration cycles is reached. The translation is estimated using

$$a_{n+1} = \frac{n}{n+1}a_n + \frac{1}{n+1}(x' - x), \tag{3.10}$$

where $a_0 = 0$. As in the centroid estimation the algorithm randomly selects points $x \in V$ and $x' \in V'$, and then computes the translation until a limit of iteration cycles is reached. We also compute the translation using

$$a_{n+1} = g'_{n+1} - g_{n+1}. \tag{3.11}$$

We show later that the centroid of the boundary of the shape approximates the real centroid in some cases. Figure 2 shows the algorithm for centroid and translation detection.

3.2 Rotation Estimation

After the algorithm computes the centroid and translation it proceeds to estimate the rotation matrix. To achieve this we first define a search area based on the centroid, and randomly select a search radius that is uniformly distributed in $r_0 \leq r \leq r_1$. Then defining the sets

$$K(r) = \{x \mid |x - g| = r\} \tag{3.12}$$

$$K'(r) = \{x' \mid |x' - g'| = r\} \tag{3.13}$$

we randomly select $x \in \partial V$ and $x' \in \partial V'$, such that

$$K(r) \cap \partial V = \emptyset \tag{3.14}$$

$$K'(r) \cap \partial V' = \emptyset. \tag{3.15}$$

We set a small threshold ϵ, for the points $x_1, x_2, x_3 \in K(r) \cap \partial V$ and the points $x'_1, x'_2, x'_3 \in K'(r) \cap \partial V'$ such that

$$||x_1 - x_2| - |x'_1 - x'_1|| \leq \epsilon \tag{3.16}$$

$$||x_2 - x_3| - |x'_2 - x'_3|| \leq \epsilon \tag{3.17}$$

$$||x_3 - x_1| - |x'_3 - x'_1|| \leq \epsilon. \tag{3.18}$$

Then we calculate the triangle vectors,

$$x_{ij} = x_i - x_j \ , \ x'_{ij} = x'_i - x'_j \tag{3.19}$$

$$x_{ij} = (x_{ij}, y_{ij})^T \ , \ x'_{ij} = (x'_{ij}, y'_{ij})^T \tag{3.20}$$

where i, j = 1,2,3. We have to ensure that the three points are not in the same line. Likewise the points are selected in the same order (see figure 3). Subsequently, the algorithm can compute the rotation matrix for each pair of triples. Since

$$x'_{ij} = Rx'_{ij} \tag{3.21}$$

we obtain

$$\begin{bmatrix} x'_{12} & x'_{23} \end{bmatrix} = R \begin{bmatrix} x_{12} & x_{23} \end{bmatrix} \tag{3.22}$$

this is equivalent to

$$\begin{bmatrix} x'_{12} & x'_{23} \\ y'_{12} & y'_{23} \end{bmatrix} = R \begin{bmatrix} x_{12} & x_{23} \\ y_{12} & y_{23} \end{bmatrix}$$

$$R = \begin{bmatrix} x'_{12} & x'_{23} \\ y'_{12} & y'_{23} \end{bmatrix} \begin{bmatrix} x_{12} & x_{23} \\ y_{12} & y_{23} \end{bmatrix}^{-1}, \tag{3.23}$$

where

$$R = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \tag{3.24}$$

and

$$r_{11} = \frac{x'_{12}y_{23} - x'_{23}y_{12}}{x_{12}y_{23} - x_{23}y_{12}} \tag{3.25}$$

$$r_{12} = \frac{x'_{23}x_{12} - x'_{12}x_{23}}{x_{12}y_{23} - x_{23}y_{12}} \tag{3.26}$$

$$r_{21} = \frac{y'_{12}y_{23} - y'_{23}y_{12}}{x_{12}y_{23} - x_{23}y_{12}} \quad (3.27)$$

$$r_{22} = \frac{y'_{23}x_{12} - y'_{12}x_{23}}{x_{12}y_{23} - x_{23}y_{12}}. \quad (3.28)$$

Then we 'vote' for r_{ij} in the accumulation space A_{ij} = (r_{ij}, score(r_{ij})). For each iteration the score(r_{ij}) = score(r_{ij}) + 1. The process of random selection of radius r, and points $\boldsymbol{x}_i$ and $\boldsymbol{x}'_i$ is repeated until the maximum number of iterations has been exceeded or a peak has been detected in the accumulation space for each r_{ij}. Figure 4 shows the procedure used to compute the rotation.

4 Experimental Results

The algorithm was tested using binary images rotated about the image centroid and then translated. The image size is 512 × 512 pixels. The centroid and translation were detected with good accuracy; the error in most cases is ± 1 pixel. For the computation it was necessary to perform sufficiently many iterations such that the law of large numbers ensures good results as shown in tables 1 and 2. In these tables the first column was estimated using eq.(3.8) and the second column using eq.(3.9).

The estimations for the translation are shown in tables 3 and 4. The first column is the actual translation applied to the figure, the second column is the estimation using eq.(3.10), the third is that using eq.(3.11) and the centroid estimated from random search eq.(3.9), and the fourth is that using eq.(3.11) and the centroid estimated from eq.(3.8). If the figure is rotated about the origin the problem is different and we must first compute the rotation and then solve the translation. For this case also we have a good approximation for the rotation. From these results we find that by using the boundary of the figure for the translation estimation we can obtain good approximations.

To compute the rotation, the algorithm searches at least one point for the same radius in both images; if a prefixed limit is reached then it looks for another radius until it can find three points. Depending on the shape and also the error allowed in matching the two triangles, the number of iterations necessary to find three points varies. The optimum value for the error of matching the two triangles is between 3 and 5 pixels. The algorithm is sensitive to the choice of this value. Likewise, for the interval for search radius, initially we must select a good estimation and limit this interval to reduce the computation time, and to ensure sufficient points to estimate the rotation matrix.

Table 5 shows the values of the rotation matrix that we can consider as good approximations, but the accuracy of these values will depend on the shape and the rotation angle, and table 6 shows the estimation of the rotation angle. This means that the accuracy of the algorithm varies. For instance, the accuracy of the algorithm using figures of regular shapes such as a square or rectangle is not perfect, but we can expect at least two of the values of the rotation matrix to be a good approximations of the rotation angle. This means that the performance of this algorithm also depends on the shape. All tests were executed on a Sun SPARC model IPX.

Table 1: *Centroid results using image boundary points*

Figure	Centroid		Estimation	
	x	y	x	y
Fig. 5	279	215	279	215
Fig. 6	219	235	219	236
Fig. 7	256	234	256	234
Fig. 8	284	222	284	223
Fig. 9	226	247	225	247
Fig. 10	259	241	259	241

Table 2: *Centroid results using all image points*

Figure	Centroid		Estimation	
	x	y	x	y
Fig. 5	281	212	281	213
Fig. 6	215	252	215	253
Fig. 7	336	222	336	221
Fig. 8	287	219	286	219
Fig. 9	222	263	222	262
Fig. 10	340	228	340	229

Table 3: *Translation results using image boundary points*

Figure	Trans.		Est. 1		Est. 2		Est. 3	
	x	y	x	y	x	y	x	y
Fig. 8	6	7	4	7	5	8	5	7
Fig. 9	8	11	6	9	6	11	7	12
Fig. 10	5	7	3	5	3	7	4	7

5 Conclusions

In this paper we introduced a basic algorithm which, by using invariant features can detect motion parameters of a planar shape without assuming point correspondences.

Table 4: *Translation results using all image points*

Figure	Trans.		Est. 1		Est. 2		Est. 3	
	x	y	x	y	x	y	x	y
Fig. 8	6	7	5	7	5	6	6	7
Fig. 9	8	11	7	11	7	9	7	11
Fig. 10	5	7	4	6	4	8	4	8

Table 5: *Rotation Matrix*

Figure	Angle	Estimation			
		r_{11}	r_{12}	r_{21}	r_{22}
Fig. 8	15.0°	0.955	0.260	-0.287	0.969
Fig. 9	45.0°	0.688	0.719	-0.712	0.712
Fig. 10	17.0°	0.968	0.280	-0.276	0.965

Table 6: *Rotation Angle*

Figure	Angle	Estimation			
		Cos	Sin	-Sin	Cos
Fig. 8	15.0°	17.13	15.09	-16.73	14.09
Fig. 9	45.0°	46.52	46.05	-45.40	44.54
Fig. 10	17.0°	14.30	16.28	-16.06	15.15

Table 7: *Time consumption*

Figure	Time (cpu unit)
Fig. 5/8	31.1u
Fig. 6/9	137.0u
Fig. 7/10	46.4u

A remaining problem is the accuracy, which is affected by (i) the number of matching points, (ii) the search radius, (iii) the triangle side error and (iv) the maximum score and number of iterations. However we can find at least three good angle approximations from the rotation matrix. The algorithm works well for smooth figures, although for angles less than 15 degrees the accuracy of some of the matrix values is poor. However in this case we can ensure that at least two values of the rotation matrix are good estimates. If we also, apply the constraint that the determinat of the rotation matrix is 1, we can select accurate estimation from these four values. Table 7 shows the algorithm performance in terms of cpu time including all processes, which also depends on the figure shape. Finally, we showed that using the image boundary is sufficient condition to estimate the motion parameters.

References

[1] Hummel, R. and Sundareswaran, V., Motion parameter estimation from global flow field data, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**, no. 5, pp. 459-476, May 1993.

[2] Taalebinezhaad, A., Direct recovery of motion and shape in the general case by fixation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14, no. 8, pp. 847-853, 1992.

[3] Kanatani, K., Tracing planar surface motion from a projection without knowing the correspondence, *Computer Vision, Graphics, and Image Processing*, **29**, pp. 1-12, 1985.

[4] Kälviäinen, H., Oja, E. and Xu, L., Randomized Hough transform applied to translational and rotational motion analysis, *Proc. 11th IAPR International Conference on Pattern Recognition*, (The Hague, The Netherlands), pp. 672-675, Aug. 1992.

[5] Haralick, R. and Zhuang, X. A note on rigid body motion from depth and optical flow, *Computer Vision, Graphics and Image Processing*, **34**, pp. 372-387, 1986.

[6] Guggenheimer H., *Differential Geometry*, Dover, New York, 1977.

[7] Kanatani, K., *Geometric Computation for Machine Vision*, Clarendon Press, Oxford, 1993.

[8] Horn, B., Closed-form solution of absolute orientation using unit quaternions, *Optical Society of America A*, 4, no. 4, pp. 629-642, April 1987.

[9] Serra J., *Image Analysis and Mathematical Morphology*, Academic Press, London, 1982.

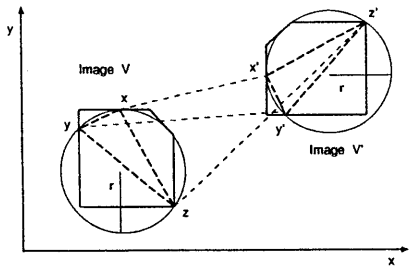

Figure 1: *Geometric description*

```
Procedure Translation – Centroid;
begin
(* Detection of centroid *)
  procedure centroid;
  begin
    while not limit do
      begin
        Select point on shape1;
        Select point on shape2;
        compute centroid shape1;
        compute centroid shape2;
      end;
  write centroid;
  end;
  (* Detection of translation *)
  procedure translation;
  begin
    while not limit do
      begin
        Select point on shape1;
        Select point on shape2;
        compute translation;
      end;
  write translation;
    end;
  end;
```

Figure 2: *Procedure for Centroid and Translation Detection*

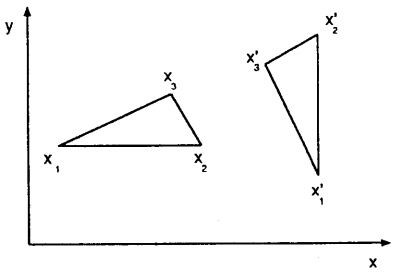

Figure 3: *Triangle order*

```
procedure Rotation;
begin
  (* Detection of rotation *)
  while not limit do
    begin
      Select triple of coplanar
      vectors from each frame;
      if (form congruent triangles)
      then begin
        compute rotation;
        vote parameter space;
        end;
    check maximum;
    end;
  write rotation;
  end; (* end procedure *)
```

Figure 4: *Procedure for Rotation Estimation*

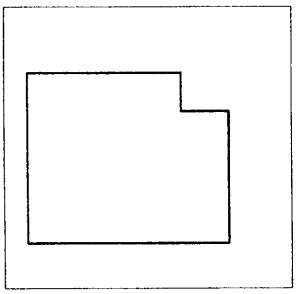

Figure 5: *Test shape 1*

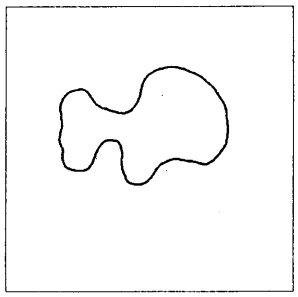

Figure 6: *Test shape 2*

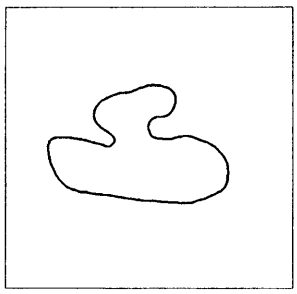

Figure 7: *Test shape 3*

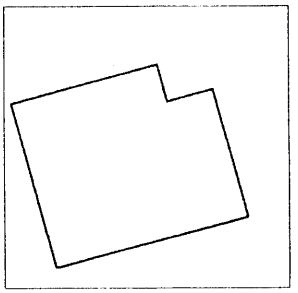

Figure 8: *Rotated and translated shape 1*

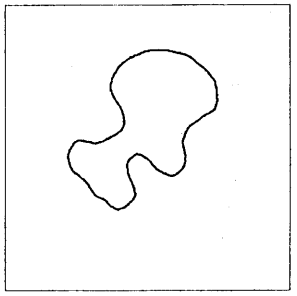

Figure 9: *Rotated and translated shape 2*

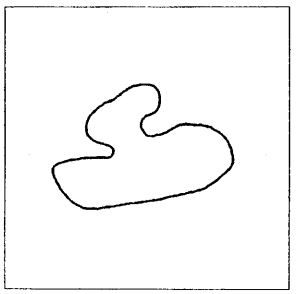

Figure 10: *Rotated and translated shape 3*