コンピュータビジョン 98－8
（1996. 1. 18）

多角形整合による確率的運動解析

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
flris@@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, 運動前と運動後との各々の図形において, 重心を中心とする円を確率的に生成し, その円と図形の境界との交点が決める多角形の頂点を対応点の候補として運動変数を計算する手法を提案する. 提案する手法は, 窓掛け確率ハフ変換の考えを運動解析に応用したものである.

キーワード 剛体運動, 運動変数, 確率算法, 対応点, 平面図形, 多角形整合

Randomized Polygon Search 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
flris@@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 motion parameters of a planar shape without knowing *a priori* the point-to-point correspondences. By randomly searching points on two shapes measured at different times, we determine the centroids, after which the algorithm proceeds to determine the rotation by randomly searching points on each shape that form congruent polygons.

Key words: rigid motion, randomized algorithm, point correspondences, planar shape, list matching, congruence

* To whom all correspondence should be addressed.

1. Introduction

The approach of this work is to use features that are invariant under Euclidean motion to detect the motion parameters of a planar shape without knowing *a priori* the point correspondences. To do this, we take two planar shapes acquired at different times by a camera moving in a static environment and then, using random search (Kälviäinen, Oja, and Xu, (1992)) and shooting circles (Agarwal and Sharir, (1993)), we compute the rotation.

The present algorithm randomly searches points on each shape to estimate the centroid of each shape. After the centroids are estimated, the rotation is computed using random selection of points in each image frame subject to the constraint that the angles between vectors constructed by three noncollinear points in the shapes are invariant. After possible point correspondences are identified, the algorithm proceeds to solve the rigid motion equation and voting for each angle 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 paper is to develop a non-model-based algorithm to estimate the motion parameters in the two-dimensional Euclidean space, which deals with the problem of point correspondences, and takes advantage of the linear constraint and the higher speed of the computation provided by the randomized approach. The algorithm uses image boundary points that provide sufficient conditions to estimate the motion parameters.

2. Rigid Object Motion

The term rigid body means an assembly of particles with fixed interparticle distances. Thus, in kinematics of solid objects, the motion of an object is rigid if and only if the distance between any two points of the body is invariant with time. Rigid motion (Guggenheimer, (1977), Kanatani, (1993)) can be described as the sum of rotation and translation 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 and y the orthogonal coordinates on R^2, we express a vector on R^2 as

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

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

$$\boldsymbol{x}' = \boldsymbol{R}\boldsymbol{x} + \boldsymbol{a} \tag{2}$$

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

$$\boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}, \quad |\boldsymbol{R}| = 1 \tag{3}$$

and the two-dimensional rotation matrix is defined as

$$\boldsymbol{R} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}. \tag{4}$$

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

$$\boldsymbol{V}' = \{\boldsymbol{x}' | \boldsymbol{x}' = \boldsymbol{R}\boldsymbol{x} + \boldsymbol{a}, \quad \boldsymbol{x} \in \boldsymbol{V}\}. \tag{5}$$

We compute the motion parameters from two shapes $\boldsymbol{V}$ and $\boldsymbol{V}'$ measured at times t_1 and t_2, respectively, and if we know three pairs of points which correspond we can solve eq.(2) and determine the motion parameters from measured images. Since the rigid motion constraint is a linear equation, we can apply the randomized Hough transform (Kälviäinen, Oja and Xu, (1992)). To develop an algorithm which estimates motion parameters without knowing *a priori* point correspondences, first, we need to examine some properties of rigid motion on a plane.

Proposition 1 *Setting $\boldsymbol{g}$ and $\boldsymbol{g}'$ to be the centroids of $\boldsymbol{V}$ and $\boldsymbol{V}'$, respectively, the relation*

$$\boldsymbol{g}' = \boldsymbol{g} + \boldsymbol{a} \tag{6}$$

holds.

Here, we set $\boldsymbol{g}=(\bar{x}, \bar{y})^T$ and $\boldsymbol{g}'=(\bar{x}', \bar{y}')^T$.

Proposition 2 *For any triplet of vectors $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z} \in \boldsymbol{V}$ we note that*
$|\boldsymbol{x} - \boldsymbol{y}|$ and $(\boldsymbol{x} - \boldsymbol{z})^T(\boldsymbol{y} - \boldsymbol{z})$ are invariant under Euclidean motion.

Theorem 1 *Let $\boldsymbol{V}$ be a planar shape; that is, $\boldsymbol{V}$ is a finite closed set on R^2. Denoting by $\partial\boldsymbol{V}$ the boundary of $\boldsymbol{V}$, for a point $\boldsymbol{g} \in \boldsymbol{V}$, we define a set*

$$\boldsymbol{K}(r) = \partial\boldsymbol{V} \cap \{\boldsymbol{x} \mid |\boldsymbol{x} - \boldsymbol{g}| = r, \quad r > 0\}. \tag{7}$$

Let $|\boldsymbol{K}(r)|$ be the number of elements of $\boldsymbol{K}(r)$, then $|\boldsymbol{K}(r)|$ is invariant under Euclidean motion.
(Proof.) For $\boldsymbol{K}(r) \subset \boldsymbol{V}$, and $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z} \in \boldsymbol{V}$ and $\boldsymbol{x}'$, $\boldsymbol{y}'$ and $\boldsymbol{z}' \in \boldsymbol{V}'$, setting

$$\boldsymbol{g}' = \boldsymbol{g} + \boldsymbol{a},$$

we obtain

$$\boldsymbol{K}'(r) = \partial\boldsymbol{V}' \cap \{\boldsymbol{x}' \mid |\boldsymbol{x}' - \boldsymbol{g}'| = r, \quad r > 0\}. \tag{8}$$

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

Theorem 1 implies that in a continuous space, we can solve the point correspondences problem by using $|\boldsymbol{K}(r)|$ and $|\boldsymbol{K}'(r)|$, since in most cases $|\boldsymbol{K}(r)|$ and $|\boldsymbol{K}'(r)|$ are finite. Thus, we select a pair of sets of polygon points from the intersection of the boundaries of the planar shapes and circles of which centers are the centroids $\boldsymbol{g}$ and $\boldsymbol{g}'$, respectively, with radius r.

3. Randomized Algorithm

For image computation we require a digital image. Hence, we proceed to detect the shape boundary by using binary dilation (Serra, (1982)). Knowledge of the boundary of the shape is sufficient to compute the motion parameters, so that we can apply the algorithm.

3.1 Estimation of Centroid

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

$$V = V_1 \cap V_2,\ V_1 \subset V,\ V_2 \subset V,\\ intV_1 \cap intV_2 = \emptyset, \tag{9}$$

where $intV$ is the interior of the region. Denoting by g, g_1, and g_2 the centroids of V, V_1, and V_2, respectively, we obtain the relation

$$g = \frac{|V_2|}{|V|}g_1 + \frac{|V_1|}{|V|}g_2, \tag{10}$$

where $|V|$ is the area measure of the set V. Thus, the centroid of a digital image can be written as

$$(\bar{x}, \bar{y})^T = \left(\frac{1}{n}\sum_{i=0}^{n} x_i,\ \frac{1}{n}\sum_{i=0}^{n} y_i\right)^T, \tag{11}$$

where $(x_i, y_i)^T$ is the centroid expressed in pixels. For our randomized approach we rewrite these equations as

$$g_{n+1} = \frac{n}{n+1}g_n + \frac{1}{n+1}x,\\ g'_{n+1} = \frac{n}{n+1}g'_n + \frac{1}{n+1}x', \tag{12}$$

for $x \in V$ and $x' \in V'$, where $g_0 = 0$, and $g'_0 = 0$.

Using a randomized approach we compute the centroid using eq.(12) that will converge after a limit of iteration cycles to the true value of the centroid of the shape. Furthermore, by proposition 1 it is clear that the translation of the shape is the translation of the centroid. The algorithm proceeds to randomly select points in $x \in \partial V$ and $x' \in \partial V'$, where ∂V and $\partial V'$ are the sets of boundary points, and compute the centroid until eq.(12) converges because the centroid of the boundary shape gives a good approximation of the centroid of the planar shape as shown in section 4.

3.2 Rotation Estimation

After the algorithm computes the centroids it then proceeds to estimate the rotation angle. To do this we first define areas for search based on the centroids, and randomly select a search radius that is uniformly distributed in $r_0 \leq r \leq r_1$. Then we define the sets of points $X = \{x_i\}_{i=1}^n$ and $X' = \{x'_i\}_{i=1}^n$, where

$$x_i \in K(r) \cap \partial V, \quad x'_i \in K'(r) \cap \partial V',$$

respectively. We determine a pair of congruent polygons of which vertices are on the circles centered at the shape centroids with radius r as is shown in figure 1. Using the property that X and X' are polygons with vertices lying on the circles, we develop an algorithm to estimate the rotation angle.

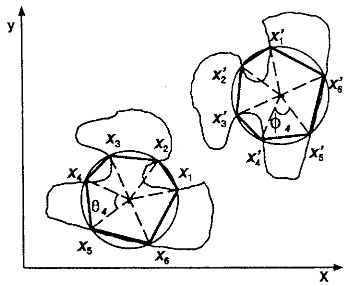

Figure 1: *Boundaries and polygons*

Let a pair of sets be $\overline{X} = \{\bar{x}_i\}_{i=1}^n$ and $\overline{X}' = \{\bar{x}'_i\}_{i=1}^n$, where

$$\bar{x}_i = x_i - g, \quad \bar{x}'_i = x'_i - g'. \tag{13}$$

Then $\overline{X}$ and $\overline{X}'$ are the polygon vertices on the circles with radius r. If x_i and x'_{i+k} correspond, we have the relation

$$\bar{x}'_{i+k} = R\bar{x}_i, \quad i = 1,2,\ldots,n, \tag{14}$$

by setting

$$\bar{x}_{i+n} = \bar{x}_i, \quad \bar{x}'_{i+n} = \bar{x}'_i, \quad i = 1,2,\ldots,n. \tag{15}$$

Thus, assuming eq.(14), we develop an algorithm using the angles between vertices by setting

$$\theta_i = \cos^{-1}\frac{\bar{x}_i^T\bar{x}_{i+1}}{||\bar{x}_i||||\bar{x}_{i+1}||},\\ \phi_i = \cos^{-1}\frac{\bar{x}'^T_i\bar{x}'_{i+1}}{||\bar{x}'_i||||\bar{x}'_{i+1}||}. \tag{16}$$

We define a pair of lists of angles

$$L(\overline{X}) = \langle\theta_1, \theta_2, \ldots, \theta_n\rangle, \quad L(\overline{X}') = \langle\phi_1, \phi_2, \ldots, \phi_n\rangle. \tag{17}$$

If eq.(14) holds, for $1 \leq j \leq n$ we obtain the relation $\theta_k = \phi_{j+k}$. Thus, we define a criterion

$$D(j) = \sum_{k=1}^{n} |\theta_k - \phi_{j+k}|. \tag{18}$$

Then a set of pairs $\{(\bar{x}_k, \bar{x}'_{i+k})\}_{k=1}^n$ which minimize $D(j)$ is a candidate for correspondences. However, eq.(18) is a sufficient condition for eq.(14). If eq.(14) holds, in continuous space we have the relation $\theta_k =$

ϕ_{j+k} for all j and k. Henceforth, we introduce a second criterion to determine the correspondences between elements of $\overline{X}$ and $\overline{X'}$. By setting

$$\gamma_k(i) = \cos^{-1} \frac{\bar{\boldsymbol{x}}_i^T \bar{\boldsymbol{x}}'_{i+k}}{||\bar{\boldsymbol{x}}_i|| ||\bar{\boldsymbol{x}}'_{i+k}||}, \qquad k = 1,2,...,n \tag{19}$$

and

$$\begin{aligned} \boldsymbol{R}_1 &= \langle \gamma_1(i), \gamma_2(i), ..., \gamma_n(i) \rangle \\ \boldsymbol{R}_2 &= \langle \gamma_n(i), \gamma_1(i), ..., \gamma_{n-1}(i) \rangle \\ \boldsymbol{R}_3 &= \langle \varepsilon_1(i), \varepsilon_2(i), ..., \varepsilon_n(i) \rangle , \end{aligned} \tag{20}$$

where $\varepsilon_k(i) = \gamma_k(i)$ - $\gamma_{k+1}(i)$ and

$$\boldsymbol{E}(i) = \sum_{k=1}^{n} \varepsilon_k(i)^2, \tag{21}$$

if $\boldsymbol{E}(i) \leq \epsilon$, for a small positive value ϵ, we adopt

$$\gamma(i) = \frac{1}{n} \sum_{k=1}^{n} \gamma_k(i) \tag{22}$$

as an estimation of the rotation angle.

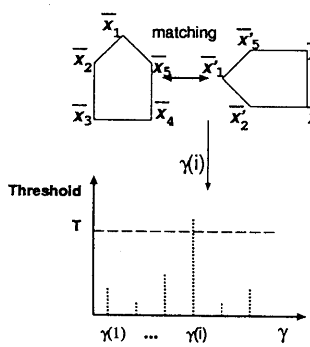

Figure 2: *Matching and voting*

From figure 2 we see that if the algorithm finds a pair of polygons which have matching $\bar{\boldsymbol{x}}_i$ and $\bar{\boldsymbol{x}}'_i$ and which minimize $\boldsymbol{D}(j)$, then we can obtain $\gamma_k(i)$. Furthermore, if $\{\gamma_k(i)\}_{k=1}^n$ minimizes eq.(21), we compute $\gamma(i)$ using eq.(22). Hence, we vote for each $\gamma(i)$ incrementing by 1 in the accumulator space. This process is repeated until a peak is detected in the accumulator space or a limit of iteration cycles is reached, then we obtain the rotation angle.

4. Experiments

The algorithm was tested using a binary image rotated around the centroid and then translated. The image size is 512 $\times$ 512 pixels.

4.1 Results

The centroid was detected with good accuracy; the error in the majority of cases is $\pm$ 1 pixel. For the computation it was necessary in this case to perform sufficient iterations such that the law of large numbers ensures good results as shown in table 1. The first column was estimated using eq.(11) and the second column using eq.(12). The translation is easily computed from this table (see table 2).

Table 1: *Centroid results*

Figure	Centroid		Estimation	
	x	y	x	y
Figure 5	350	135	350	134
Figure 6	355	141	355	141
Figure 7	238	262	238	263
Figure 8	241	265	240	265

Table 2: *Translation using the centroid*

Figure	Real		Estimation	
	x	y	x	y
Figure 5/6	6	7	5	7
Figure 7/8	3	3	2	2

To compute the rotation, we generate circles C with radius r of which center is g and circles C' with the same radius r of which center is g'. The common points of ∂V and C and $\partial V'$ and C' determine a pair of congruent convex polygons. We generate these circles from an appropiate interval $r_o \leq r \leq r_1$.

The interval for searching radius r affects the computation time, and the number of circles affects the results of the estimated rotation angle. Thus, we must select a good estimation of the interval for searching radius r and we need a sufficient number of points for each polygon. Table 3 shows the intervals, threshold, and the number of iterations for shapes 1 and 2.

Table 4 shows two estimations for the rotation angle of which error is $\pm$ 1.0 degree. The performance of the algorithm is shown in table 5 and it depends on the number of polygon vertices and the shape.

Table 3: *Thresholds*

Shape	r_0	r_1	Threshold	Iterations
Shape 1	99	144	15	1500
Shape 2	92	137	15	2500

Table 4: *Rotation Angle*

Figure	Rotation	Est. 1	Est. 2
Figure 5/6	15.0°	14.85°	13.63°
Figure 7/8	23.0°	23.22°	23.97°

Table 5: *Time consumption*

Figure	Time (s.)
Figure 5/6	20.8
Figure 7/8	45.0

4.2 Error Analysis

The majority of the error is due to the image digitalization although roundoff error also contributes. Thus, we analyse the error caused by digitalization. The digitalization affects location of the centroids, translation and boundary detection.

If location of the centroids and translation are correct but there is inaccuracy in the boundary detection, then as shown in figure 3, a pair of corresponding points ($\boldsymbol{x}$, $\boldsymbol{x}'$) changes to ($\boldsymbol{x}^*$, $\boldsymbol{x}'^*$). By assuming $|\boldsymbol{x} - \boldsymbol{x}^*| = \delta$, $|\boldsymbol{x}' - \boldsymbol{x}'^*| = \delta$, and that the radii of the circles are r, we obtain

$$\tan|\omega' - \omega| = \frac{\delta}{r}. \tag{23}$$

This leads to

$$\Delta\omega = \frac{\delta}{r}\frac{180}{\pi}, \quad \Delta\omega = |\omega' - \omega|. \tag{24}$$

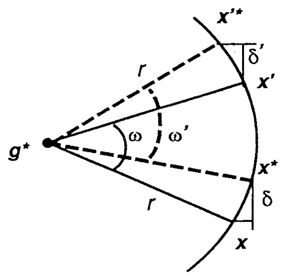

Figure 3: *Digitalization error*

Additionally, if the boundaries are correct but there are inaccuracies in the estimation of the location of the centroids and translation, we obtain the point configuration shown in figure 4. Let r and $r' = r + \epsilon$ be the radius of the circles whose centers are $\boldsymbol{g}$ and $\boldsymbol{g}^*$, and $\epsilon \ll 0$. Then we have $r' \approx r$. The difference of the rotation angles which are estimated using $\boldsymbol{g}$ and $\boldsymbol{g}^*$ is $|\theta' - \theta|$. As in the case of $\Delta\omega$ we obtain

$$\Delta\theta = \frac{\delta}{r}\frac{180}{\pi}, \quad \Delta\theta = |\theta' - \theta|. \tag{25}$$

Setting the total error of the estimation of the rotation angle to be E, E is approximately expressed as

$$E \approx \Delta\theta + \Delta\omega \approx 2\max(\Delta\theta, \Delta\omega), \tag{26}$$

since $\Delta\theta$ and $\Delta\omega$ are approximately independent. Setting $\delta = 1$ pixel and the average of r to be 100 pixels, we obtain $E \approx 2°$. This corresponds to the error of our experimental results.

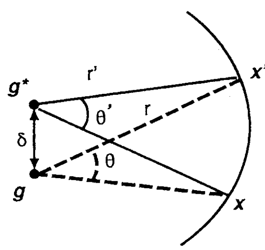

Figure 4: *Rotation error*

By increasing the resolution of the digitalization, r increases and δ decreases for any figure. This indicates that we can improve the accuracy of the estimation of the rotation angle if we obtain a higher resolution of the digital images.

5. Conclusions

We showed that we can estimate the motion parameters using a randomized approach, without knowing *a priori* the point correspondences. The remaining problems are the accuracy that depends on the image digitalization and the performance of the algorithm which can be improved using a parallel algorithm.

The proposed algorithm is categorized into window RHT (Kälviäinen, Hirvonen, Xu, and Oja, (1995)), since we can consider circles shooting for polygon search as a window procedure for sample points.

References

Agarwal, P. and Sharir M., (1993), Circle Shooting in a Simple Polygon, *Journal of Algorithms*, 14, pp. 64-87.

Guggenheimer H., (1977), *Differential Geometry*, Dover, New York.

Kälviäinen, H., Oja, E. and Xu, L., (1992), 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.

Kälviäinen, H., Hirvonen, P., Xu, L., and Oja, E., (1995), Probabilistic and Non-probabilistic Hough Transform: Overview and Comparisons, *Image and Vision Computing*, **13**, pp. 239-252.
Kanatani, K., (1993), *Geometric Computation for Machine Vision*, Oxford University Press, Oxford.
Serra J., (1982), *Image Analysis and Mathematical Morphology*, Academic Press, London.

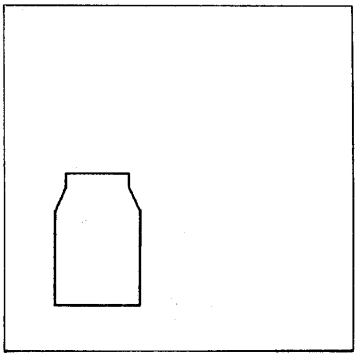

Figure 5: *Test shape 1*

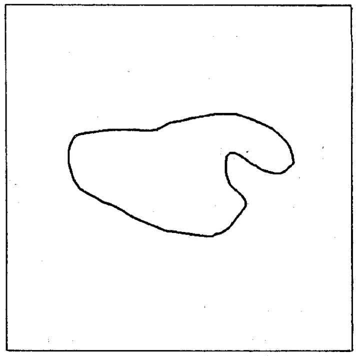

Figure 7: *Test shape 2*

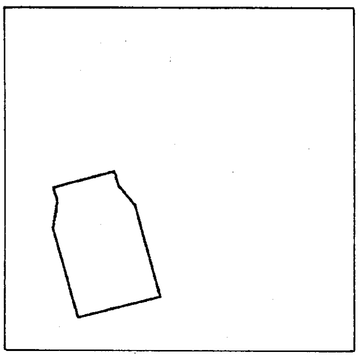

Figure 6: *Rotated and translated shape 1*

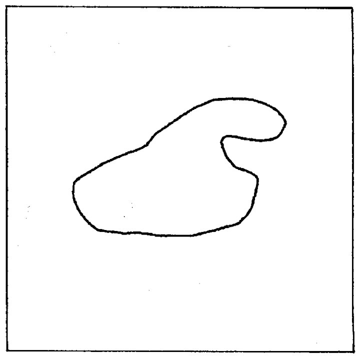

Figure 8: *Rotated and translated shape 2*