An Extension of the Generalized Hough Transform to Realize Affine-Invariant Two-dimensional(2D) Shape Detection

Akio KIMURA and Takashi WATANABE
Department of Computer and Information Science, Iwate University
E-mail: kimura@cis.iwate-u.ac.jp, watanabe@cis.iwate-u.ac.jp

Abstract

In this paper, we present a new method for two-dimensional (2D) shape detection applicable under affine transformation. The problem of affine-invariant shape detection is an important and fundamental research subject in computer vision. Although various methods have been proposed to solve this problem, most of those approaches are not well suited for the following general cases: (1) a shape to be detected is occluded by other overlapping objects, (2) a shape boundary is partially broken because of noise or other factors. We introduce a new method to deal with such cases, which extends the generalized Hough transform [1] to be an affine-invariant shape detector. This method, called the affine-GHT, utilizes pairwise parallel tangents and basic properties of an affine transformation to carry the direct computation for six parameters of an affine transformation. Experimental result demonstrates that the proposed method performs successfully and efficiently.

1. Introduction

In many industrial applications of image pattern recognition, it is a significant research subject how to establish an affine-invariant 2D shape detection method. This problem is also important in the three-dimensional(3D) image analysis such as computer vision. Because it is well known that under the paraperspective projection of the scene, two different images of the same flat object are in an affine 2D correspondence [2].

Many studies have been produced for solving such a problem [2–20]. One approach is a model- or appearance-based technique (such as the geometric hashing [2–4], the invariant pixel set signature [7] or similar other methods). In this type of methods, a shape is represented by invariant features or signatures to affine transformation, and a correlation- or statistical-based matching using those is applied to detect shapes. Several kinds of invariants have been devised and utilized, e.g. based on local or global features [8–10], based on numerical or statistical analysis [11–13, 16], etc. Moreover, another approaches to affine-invariant shape detection have also been explored recently: using genetic algorithm [18], using neural networks [19], using wavelet transform [20], and so on.

For the above conventional methods, we can see that an efficient shape detection is possible under the certain restricted or constrained environments, e.g., where many attractive or interest features may be easily detected or the shape's pose and illumination are tightly controlled. However most of them are not well suited for the following general cases: (1) a shape to be detected is occluded by other overlapping objects, (2) a shape boundary is partially broken or cluttered because of noise or other factors, (3) a shape to be detected is composed of general curves, and (4) the detection process needs to be executed in a real-time. Therefore, it is desired to develop a new, efficient and powerful shape detection method that can deal with such cases.

On the other hand, if we focus on the similarity-invariant (translation-, rotation-, and scale-invariant) shape detection, the generalized Hough transform (GHT [1]) is well known as a robust method and it is also a time consuming. It should be noted that the GHT may be extended as a powerful affine-invariant shape detector through further improvements.

In this paper, we extend the GHT to be as an affine-invariant shape detector. This method, called *the affine-GHT*, positively utilizes pairwise parallel tangents and basic properties of an affine transformation. The affine-GHT is available even thoughthe conventional shape detectors seem to be difficult to treat.

2. Proposed method (affine-GHT)

As a 2D affine transformation is described with six parameters (see below), we need a six-dimensional(6D) voting space to directly apply a method based on the Hough transform. However it seems impossible in actuality to execute the method.

In this paper, we introduce a multi-stage Hough transform. At the first stage, we apply a 2D Hough transform

1051-4651/02 $17.00 © 2002 IEEE

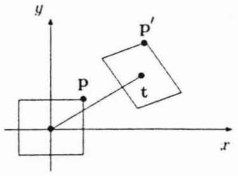

Figure 1. An affine transformation.

to obtain candidate positions of a target shape, and at the second stage, apply four-dimensional(4D) Hough transform to obtain the remained four parameters. Since we need to search a 4D voting space at the second stage, we adopt the adaptive Hough transform (AHT [22]) to improve the processing efficiency, which executes vote processing by changing adaptively the resolution of a voting space.

In the following, our discussion is made on a boundary edge image that contains tangent information at every edge point. (In this paper, all tangent informations were obtained by using an improved method of [21].)

2.1. Definition of affine transformation

Consider two orthogonal coordinate systems, Σ and Σ'. Let $\mathbf{p} = (x, y)^t$ and $\mathbf{p}' = (x', y')^t$ be, respectively, the coordinates of a point P on the Σ and the Σ'. An affine transformation is defined as follows: $\mathbf{p}$ is transformed into $\mathbf{p}'$ by

$$\mathbf{p}' = A\mathbf{p} + \mathbf{t}, \tag{1}$$

where $\mathbf{t}$ is a translation vector denoted by $\mathbf{t} = (t_x, t_y)^t$, A is a linear transformation matrix denoted by $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, and $ad - bc \neq 0$ (see Fig.1).

Eq.(1) shows that the affine transformation has six parameters. Therefore, if three corresponding pairs of $\mathbf{p_i}$ and $\mathbf{p_i}'$ $(i = 1, 2, 3)$ are given, all the affine-parameters $\{a,b,c,d,t_x,t_y\}$ are uniquely determined.

The affine transformation has important properties.

- A pair of parallel straight lines is transformed into also a pair of parallel straight lines after transformation (preserving parallelism).
- The centroid of the shape is transformed into also the centroid of the transformed shape.

We will positively utilize these properties in the following algorithm.

2.2. Details of algorithm

We call a target 2D shape to be detected *a template* T. The boundary of T is described by a set of N edge points expressed as $\{(X_i, Y_i), i = 1, 2, \cdots, N\}$. Here, the origin of the XY-coordinate system is put to the reference point R that may be arbitrarily set on T.

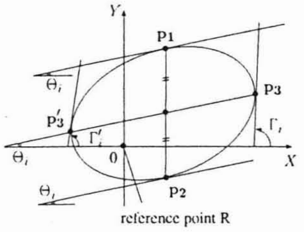

Figure 2. Description of a template T.

2.2.1 Construction of R-Table and E-Table

First, we generate a R-table (reference-table, like GHT [1]) to represent a template T by the following procedure.

Procedure *R-table construction* (see Fig.2)

1. Prepare a 1D array of linked lists as shown in Table 1 to store all parameters described below. The index of the array is the tangent value, Θ.
2. All edge pixels of T are grouped and sorted according to their tangent values Θ.
3. Select a pair of two edge points $\mathbf{p_1} = (X_1, Y_1)^t_i$ and $\mathbf{p_2} = (X_2, Y_2)^t_i$ that have the same tangent Θ_i.
4. Draw a straight line through the midpoint of the segment $\overline{\mathbf{p_1 p_2}}$ to the direction Θ_i, and find all edge points that the line intersects T again. Let these points be, respectively, $\mathbf{p_3} = (X_3, Y_3)^t_i$, $\mathbf{p_3}' = (X_3', Y_3')^t_i$, $\cdots$, and their tangents be, respectively, Γ_i,Γ_i',$\cdots$.
5. Register the parameters Θ_i, $(X_1, Y_1)_i$, $(X_2, Y_2)_i$, $(X_3, Y_3)_i$, $(X_3', Y_3')_i$, $\cdots$, Γ_i, Γ_i', $\cdots$ into the R-table (see Table 1).
6. Repeat the previous step 2 $\sim$ step 5 for all possible $\mathbf{p_1}$,$\mathbf{p_2}$, and Θ_i.

Second, we generate an E-table (edge-table) from an input image I by the following procedure.

Procedure *E-table construction* (see Fig.3)

1. Prepare a 1D array of linked lists as shown in Table 2. The index of this array is the tangent value, θ.
2. All edge pixels on I are grouped and sorted according to their tangent values θ.
3. Select a pair of two edge points $\mathbf{q_1} = (x_1, y_1)^t_j$ and $\mathbf{q_2} = (x_2, y_2)^t_j$ that have the same tangent θ_j.
4. Register the parameters θ_j, $(x_1, y_1)_j$, and $(x_2, y_2)_j$ into the E-table (see Table 2).
5. Repeat the previous step 3 $\sim$ step 4 for all possible $\mathbf{q_1}$,$\mathbf{q_2}$, and θ_j.

2.2.2 Detection of translation t

At the first stage of vote processing, we detect candidate positions of the reference point R of T.

Procedure *translation detection*

1. Repeat the following step 2 $\sim$ step 5 for $\theta = 0$ to π with an increment $\delta\theta$.

Table 1. Contents of the R-table.

tangent value	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3, \mathbf{p}_3', \cdots$	tangents at the points $\mathbf{p}_3, \mathbf{p}_3', \cdots$
Θ_i	$(X_1, Y_1)_i$	$(X_2, Y_2)_i$	$(X_3, Y_3)_i, (X_3', Y_3')_i, \cdots$	$\Gamma_i, \Gamma_i', \cdots$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

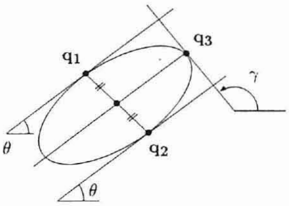

Figure 3. Construction and utilization of the E-table from an input image.

2. Select a record with $\theta = \theta_j$ from the E-table, and let $\mathbf{q}_1$ and $\mathbf{q}_2$ be $\mathbf{q}_1 = (x_1, y_1)^t$ and $\mathbf{q}_2 = (x_2, y_2)^t$, respectively. When no such records in the E-table, back to step 1.
3. Draw a straight line through the midpoint of the segment $\overline{\mathbf{q}_1\mathbf{q}_2}$ to the direction θ on I (see Fig.3), and find an edge point $\mathbf{q}_3 = (x_3, y_3)^t$ which exists on this line. Let the tangent value at $\mathbf{q}_3$ be γ.
4. For all records in the R-table, repeat the following step (a) ∼ step (e).
   (a) For a record Θ_i, let Θ,$\mathbf{p}_1$,$\mathbf{p}_2$,$\mathbf{p}_3$, and Γ be, respectively, $\Theta = \Theta_i$, $\mathbf{p}_1 = (X_1, Y_1)_i^t$, $\mathbf{p}_2 = (X_2, Y_2)_i^t$, $\mathbf{p}_3 = (X_3, Y_3)_i^t$ and $\Gamma = \Gamma_i$.
   (b) Calculate six affine parameters $\{a,b,c,d,t_x,t_y\}$ by using three corresponding pairs of edge points ($\mathbf{p}_j \leftrightarrow \mathbf{q}_j$, $j = 1, 2, 3$) and their relationships

$$\mathbf{q}_j = A\mathbf{p}_j + \mathbf{t}. \qquad (2)$$

   (c) Calculate (x_θ, y_θ) and (x_γ, y_γ) by

$$\begin{pmatrix} x_\theta \\ y_\theta \end{pmatrix} = A \begin{pmatrix} \cos\Theta \\ \sin\Theta \end{pmatrix}, \quad \begin{pmatrix} x_\gamma \\ y_\gamma \end{pmatrix} = A \begin{pmatrix} \cos\Gamma \\ \sin\Gamma \end{pmatrix}. \qquad (3)$$

   (d) For allowable angle-errors Δ_1 and Δ_2, if

$$|\theta - \arctan(y_\theta/x_\theta)| \le \Delta_1 \qquad (4)$$

$$|\gamma - \arctan(y_\gamma/x_\gamma)| \le \Delta_2 \qquad (5)$$

   are both satisfied, add 1's to the element $H_{xy}(t_x, t_y)$ of a 2D voting array H_{xy} whose size is the same to I.
   (e) Repeat the previous step (b) ∼ step (d) as to $\mathbf{p}_3 = (X_3', Y_3')_i^t$ and $\Gamma = \Gamma_i'$. (Moreover, if $(X_3'', Y_3'')^t$, Γ'', $\cdots$ are also registered in the R-table, repeat the same procedure as to each of them.)

Table 2. Contents of the E-table.

tangent value	edge point $\mathbf{q}$
θ_j	$(x_1, y_1)_j, (x_2, y_2)_j, \cdots$
$\cdots$	$\cdots$

5. If another edge point $\mathbf{q}_3'$ is found in step 3, repeat the previous step 4 as to it. If not found, back to step 2.
6. After all voting for possible θ's are finished, extract all peak elements $H_{xy}(t_{kx}, t_{ky})$ whose voting values are over a threshold $\mathrm{TH_t}$ as candidate positions of T. Let the set of these peaks be $\{\mathbf{t_k} = (t_{kx}, t_{ky})^t\}$.

2.2.3 Detection of linear transformation A

For each $\mathbf{t_k}$, we calculate a linear transformation parameter $A_k = \{a_k,b_k,c_k,d_k\}$. However, four parameters $\{a_k,b_k,c_k,d_k\}$ are all real numbers and their changing ranges are respectively different. Therefore, we introduce an approach like the AHT [22] to reduce memory requirement and improve the efficiency of voting procedures.

Procedure *linear transformation detection*

7. Prepare a list L_k for each candidate $\mathbf{t_k}$. Repeat the previous step 1 ∼ step 6 again and register parameters $\{a_{ki},b_{ki},c_{ki},d_{ki}\}$ into the list L_k whenever a voting to the element $H_{xy}(t_{kx}, t_{ky})$ is carried out.
8. Prepare a four-dimensional array H_{abcd} to obtain linear transformation parameters $\{a,b,c,d\}$ whose size is $9 \times 9 \times 9 \times 9$. For each record registered in the list L_k, repeat the following step (a) ∼ step (e).
   (a) Determine the maximum fluctuation ranges of four parameters $\{a_{ki},b_{ki},c_{ki},d_{ki}\}$ by scanning the list L_k. The whole range of the voting space H_{abcd} is set to their four maximum ranges.
   (b) Add 1's to the element $H_{abcd}(a_{ki}, b_{ki}, c_{ki}, d_{ki})$ corresponding to each record registered in the list L_k. After voting for all records is finished, extract all peak elements of the array H_{abcd} whose voting values are over a threshold $\mathrm{TH_A}$. Push these peaks $A_k = \{a_k,b_k,c_k,d_k\}$ into a stack.
   (c) Pop a peak element A_k from the stack. If the stack is empty, proceed to step 9.
   (d) If each resolution of the parameters $\{a_k, b_k, c_k, d_k\}$ is enough (for example, its ratio to the initial less than 0.01), register $\mathbf{t_k}$ and

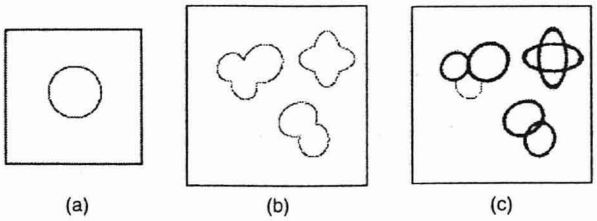

Figure 4. Detection of multiple broken ellipses: (a) the circle template of 158×158 pixels, (b) the tested ellipses of 256×256 pixels, (c) the detected ellipses (drawn by the thick line).

A_k into a list ANS to store solutions, and back to step (c). Otherwise, proceed to the next.

(e) Considering the range of $3 \times 3 \times 3 \times 3$ elements centered on A_k to be the whole space of H_{abcd}, divide it into $9 \times 9 \times 9 \times 9$ elements again. However, the parameter axis whose resolution has already reached an adequate value is not divided.

9. Repeat the previous step 8 for all lists $\{L_k\}$. All records in the list ANS are final solution parameters.

3. Experiments and discussion

In order to verify the validity and the effectiveness of the proposed algorithm, we carried out evaluation experiments. We used a Pentium II processor (400MHz) to compute. The following parameters were used in setting up the experiments: $\delta\theta = 1°$, $\Delta_1 = \Delta_2 = 5°$, $\mathrm{TH_t} = 500$, and $\mathrm{TH_A} = 30$.

First, we tried to detect multiple broken ellipses using a circle template (see Fig.4). Note that an affine-transformed circle is to be an ellipse. The result is shown in Fig.4(c). Processing time was 12.3 seconds. As shown in the result, the detected ellipses are quite well, especially the concentric ellipses are correctly separated and detected. Although only one ellipse was not detected (which is to the left-upper corner, one of the overlapped three ellipses), this is due to lack of parallel tangentsfor this partial ellipse. It is necessary to keep in mind that the proposed method may not work well when a boundary of the shape is continuously broken more than 50% or so.

Second, we tested real images of flat industrial parts and tried to detect each part. As space is limited, only three experimental results are shown in Fig.5. In these figures, columns (a) and (c) are real images taken through a digital camera, and columns (b) and (d) are the boundary edge images obtained by the Canny filter [23], respectively. Note that each part to be detected in the tested images has undergone an affine transformation, since each camera viewpoint is different. In addition, the boundary of the target part is partially cluttered and occluded by other overlapping objects. The detected results are shown in column (e) of Fig.5. Their processing times were 11.2, 16.4 and 5.3 seconds, in order from the top. The upper and middle results demonstrate that each detected shape is quite close to the original shape and the proposed method works correctly in the practical environments. The lower images of Fig.5 are listed to show an example of failure detection. As shown in this example, when an original gray image is strongly blurred, we cannot obtain valid tangents at edge points in such images. Thus a more robust method of extracting tangents from a boundary image will be needed to apply the proposed method more widely. However, on the whole we can confirm that the affine-GHT performs successfully and efficiently in real images.

While the proposed method can treat with the practical case that a shape boundary is partially occluded or cluttered, its memory requirement is small and its processing is fast. Furthermore, it is important that the Ballard's GHT can be extended as an affine-invariant shape detector. In particular, although the Ballard's GHT needs a brute force voting to determine two parameters of rotation and scale, our method does "not" use a brute force voting to determine four parameters $\{a,b,c,d\}$ of linear transformation.

Future works of the proposed method are in further improvements of the processing efficiency and the detection robustness. For example, the processing time will increase if a shape to be detected has a large number of edge pairs whose tangents are parallel (like rectangle, rhombus, etc.), because its R-table becomes enormous. In such a case, we need to improve the algorithm so that it may adopt a voting form based on geometric features such as line segments, rather than edge points.

4. Conclusion

An efficient method to realize the affine-invariant 2D shape detector, called *the affine-GHT*, is presented. The affine-GHT is available to detect arbitrary 2D shapes robustly even though a shape boundary is partially occluded or broken. The affine-GHT intends to detect 2D shapes mainly, but it will be applicable to 3D image analysis as described in the beginning of this paper.

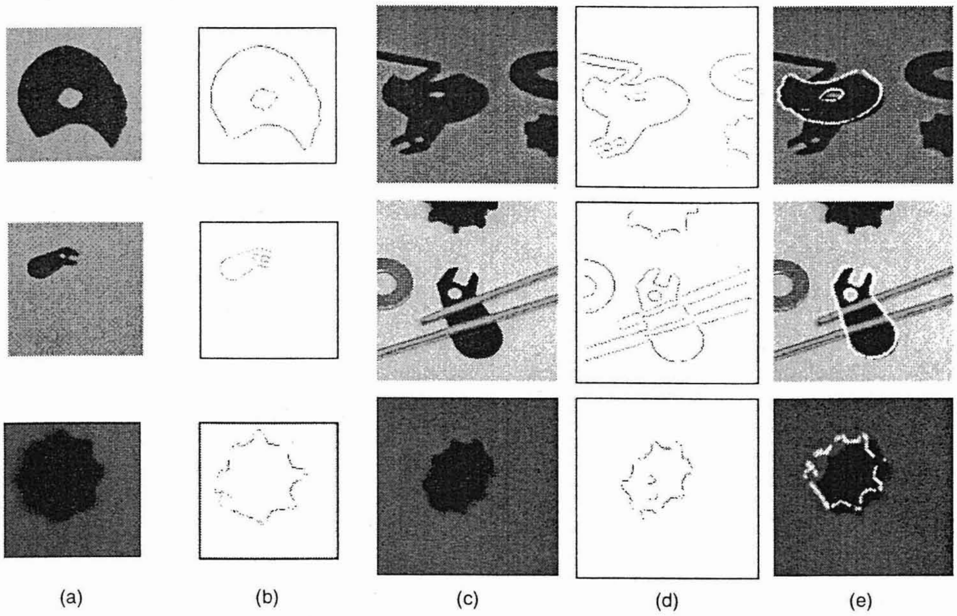

Figure 5. Experimental results for real images: for each column, (a) the template image of 158×158 pixels, (b) the edge image of the template, (c) the tested image of 256×256 pixels, (d) the edge image of the tested image, (e) the detected result (the affine-transformed boundary of the template is overwritten).

Acknowledgement

This work was partially supported by the National Grand-in-Aid for Scientific Research, No.14780253.

References

[1] D.H.Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[2] Y.Lamdan, J.T.Schwartz, and H.J.Wolfson. Affine invariant model-based object recognition. *IEEE Trans. Rob. Aut.*, 6:578–589, 1990.

[3] M.S.Costa, R.M.Haralick, and L.G.Shapiro. Optimal affine-invariant matching: Performance characterization. *SPIE*, 1662:21–34, 1992.

[4] F.C.D.Tsai. A probabilistic approach to geometric hashing using line features. *Computer Vision and Image Understanding*, 63(1):182–195, 1996.

[5] J. Ben-Arie and Z.Wang. Pictorial recognition of objects employing affine invariance in the frequency domain. *IEEE Trans. Pattern Anal. & Mach. Intell.*, 20(6):604–618, 1998.

[6] H.G.Feichtinger and T.Strohmer. *Gabor Analysis and algorithms –theory and applications–*. Birkhauser Verlag, 1997.

[7] J.Matas, J.Burianek, and J.Kittler. Object recognition using the invariant pixel-set signature. *Proc. of the British Machine Vision Conference*, September 2000.

[8] D.G.Lowe. Object recognition from local scale-invariant features. *Proc. of the International Conference on Computer Vision*, Corfu, 1150–1157, 2000.

[9] Schmid C. and R.Mohr. Local grayvalue invariants for image retrieval. *IEEE Trans. Pattern Anal. & Mach. Intell.*,19(5):530–534, 1997.

[10] A.M.Bruckstein, R.J.Holt, A.Netravali, and T.J.Richardson. Invariant signatures for planar shape recognition under partial occlusion. *Computer Vision, Graphics and Image Processing*, 58(1):49–65, 1993.

[11] E.Calabi, P.J.Olver, C.Shakiban, A.Tannenbaum, and S.Haker. Differential and numerically invariant signature curves applied to object recognition. *Int.Journal of Computer Vision*, 26:107–135, 1998.

[12] R.Berthilsson and A.Heyden. Recognition of planar objects using the density of affine-shape partial occlusion. *Computer Vision, Graphics and Image Processing*, 58(1):49–65, 1993.

[13] J.Subrahmonia, D.Cooper, and D.Keren. Practical reliable Bayesian recognition of 2D and 3D objects using implicit polynominals and algebraic invariants. *IEEE Trans. Pattern Anal. & Mach. Intell.*,18:505–519, 1996.

[14] M.Werman and D.Weinshall. Similarity and affine invariant distances between 2D point sets. *IEEE Trans. Pattern Anal. & Mach. Intell.*,17(8):810–814, 1995.

[15] S.Startchik, R.Milanese, C.Rauber, and T.Pun. Planar shape databases with affine invariant search. *Proc. of the International Workshop on Image Databases and Multi Media Search*, August 1996.

[16] D. Zhao and J. Chen. Affine curve moment invariants for shape recognition. *Pattern Recognition*, 30(6):895–901, 1997.

[17] D.Shen, Horace H.S.Ip, and E.K.Teoh. Affine invariant detection of perceptually parallel 3D planar curves. *Pattern Recognition*, 33:1909–1918, 2000.

[18] S.Y.Yuen and C.H.Ma. Genetic algorithm with competitive image labelling and least square. *Pattern Recognition*, 33:1949–1966, 2000.

[19] G. Bebis, M.Georgiopoulos, N.V.Lobo, and M.Shah. Learning affine transformations. *Pattern Recognition*, 32(10):1783–1799, 1999.

[20] M.I.Khalil and M.M.Bayoumi. A dyadic wavelet affine invariant function for 2D shape recognition. *IEEE Trans. Pattern Anal. &Mach. Intell.*,23(10):1152–1164, 2001.

[21] T. Watanabe, M. Hatakeyama, and A. Kimura. Extraction of tangent information and detection of broken ellipses using hough transform. *Trans. of the Institute of Electronics, Information and Communication Engineers*, J82-D-II(12):2221–2229, December 1999, *in Japanese*

[22] J.Illingwoth and J. Kittler. The adaptive Hough transform. *IEEE Trans. Pattern Anal. & Mach. Intell.*, 9(5):690–698, 1987.

[23] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. & Mach. Intell.*, 8(6):679–698, 1986.