

Image normalization for pattern recognition

Soo-Chang Pei and Chao-Nan Lin

In general, there are four basic forms of distortion in the recognition of planar patterns: translation, rotation, scaling and skew. In this paper, a normalization algorithm has been developed which transforms pattern into its normal form such that it is invariant to translation, rotation, scaling and skew. After normalization, the recognition can be performed by a simple matching method. In the algorithm, we first compute the covariance matrix of a given pattern. Then we rotate the pattern according to the eigenvectors of the covariance matrix, and scale the pattern along the two eigenvectors according to the eigenvalues to bring the pattern to its most compact form. After the process, the pattern is invariant to translation, scaling and skew. Only the rotation problem remains unsolved. By applying the tensor theory, we find a rotation angle which can make the pattern invariant to rotation. Thus, the resulting pattern is invariant to translation, rotation, scaling and skew. The planar image used in this algorithm may be curved, shaped, a grey-level image or a coloured image, so its applications are wide, including recognition problems about curve, shape, grey-level and coloured patterns. The technique suggested in this paper is easy, does not need much computation, and can serve as a pre-processing step in computer vision applications.

Keywords: affine transform, image normalization, pattern recognition

INTRODUCTION

Pattern recognition has been an important area in computer vision applications. In the case of a planar image, there are four basic forms of geometric distortion caused by the change in camera location: translation, rotation, scaling and skew. So far, a number of methods have been developed to solve these distortions, such as moment invariants¹, Fourier descriptor^{2,3}, Hough transformation⁴, shape matrix⁵ and the principle axis method⁶. All of the above methods can

be made invariant to translation, rotation and scaling. However, they become useless when pattern is skewed: when the direction of the camera is not vertical to the planar image or the sampling intervals in the $x - y$ directions are not equal, the image is skewed.

A tensor-based moment function method has been developed to recognize objects under distortion of translation, rotation, scaling and skew⁷. The method applies the tensor-based moment function to compute the affine transformation between the two images according to the inverse of the calculated affine transformation. No knowledge of point correspondence between images is required, but the method needs to compute fourth-order moments; however, we know that higher-order moments are more sensitive to noise and digitization effects⁸. Another defect is that the method is not efficient when patterns are large, as the method must compute the affine transformation between the input pattern and each of the reference patterns until the pattern is identified. In this paper, we introduce the concept of image normalization, which normalizes all of the images before recognition. Thus, we just compare the input normalized pattern with the reference patterns using a matching method, which is very simple and fast.

A block diagram of pattern recognition by image normalization is shown in *Figure 1*. This method first extracts features from the input patterns, then normalizes the input pattern with a normalization algorithm. Here, we define normalization as a process which transforms the input pattern into a normal form that is invariant under translation, rotation, scaling and skew. We call the transformed image a *normalized image*. Since the normalized image is invariant under translation, rotation, scaling and skew, we can recognize patterns just by a simple matching method.

A good method proposed by Leu⁹, can solve skew

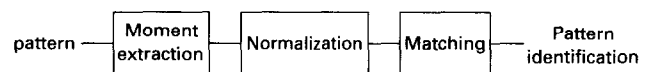


Figure 1 Pattern recognition by normalization and matching.

Institute of Electrical Engineering, National Taiwan University, Taipei, Taiwan, ROC
Paper received: 16 February 1993; revised paper received: 12 January 1995

distortion efficiently. It first rotates the pattern according to the eigenvectors of the covariance matrix, then rescales the pattern along the eigenvectors according to the corresponding eigenvalues. The resulting pattern becomes compact, which we call the *compact image*. We shall prove that only the rotation problem remains unsolved through the process. In this paper, we extend the method to achieve rotation invariance. By the tensor theory, we derive a simple equation from which an angle can be calculated to make the pattern invariant to rotation. Thus, the resulting pattern is invariant to translation, rotation, scaling and skew.

Over the years, a large number of existing recognition methods have only made use of the geometrical properties among various patterns. In fact, the information inside patterns is also important in pattern recognition. For example, as we show in *Figure 2a*, these two patterns are in the shape of a square, but their content is different, so they should be classified as different patterns. Unfortunately, most of existing algorithms cannot distinguish one from the other.

Another method, proposed by Ming Fang and Gerd Hansler¹⁰, can solve the problem of different contents. Their algorithm is based on a general transformation where the kernel itself contains the function to be transformed; thus, the invariance is achieved by a kind of self-mapping. In other words, their algorithm utilizes the contents of patterns instead of the geometrical characteristics of patterns. The transform is a many-to-few mapping; it can happen that two patterns have the same mapping. Their method fails when the patterns have the same contents but the relative positions of the component parts are different. For example, as in *Figure 2b*, the shapes and contents of these two patterns are the same, but the relative positions of the component parts in each pattern are different. Thus,

they should be classified as different patterns. In this paper, we utilize information on both shape and content to recognize a pattern and to solve all the problems mentioned above.

In the next section we show how to extract features which contain information on geometry and content simultaneously. We then review the compact algorithm analyse the properties of the compact image, apply the tensor theory to a normalized image, and summarize the normalization algorithm. Finally, we draw conclusions, which discuss how efficient the matching method performs using image normalization.

To unify the syntactic notation we use bold lower case letters to denote a vector, bold upper case to denote a matrix and text to denote a variable.

FEATURE EXTRACTION

In this section we introduce a moment representation that contains both the geometrical and internal information of patterns. We first show how to define the moments which contain the geometrical and internal information simultaneously. Then, we derive a theorem which is important in this paper.

Definition Let $p(x, y)$ denote the image signature at location (x, y) . For example:

$$p(x, y) = \begin{cases} 1: \text{object, 0: background} & \text{for curve or shape} \\ \text{grey-level value} & \text{for grey image} \\ \text{brightness or chrominance} & \text{for coloured image} \end{cases} \quad (1)$$

Denote the probability density function by $f(x, y)$:

$$f(x, y) = \frac{p(x, y)}{\int_{\Omega} p(x, y) dx dy} \quad (2)$$

where Ω is the region we consider.

Hence, we can calculate the mean of the image; let $c = [C_x, C_y]^T$ denote the mean vector:

$$C_x = \int_{\Omega} x f(x, y) dx dy \quad (3)$$

$$C_y = \int_{\Omega} y f(x, y) dx dy$$

Let u_{kr} denote the central moments of $k + r$ order:

$$u_{kr} = E\{(X - C_x)^k (Y - C_y)^r\} \quad (4)$$

$$= \int_{\Omega} (x - C_x)^k (y - C_y)^r f(x, y) dx dy$$

Since $f(x, y)$ contains the internal information of patterns, the moments defined above contain both geometrical and internal information on the patterns.

Properties

Consider a pattern which is distorted by the change of a camera's location. the distortion may be translation, rotation, scaling or skew, all of which are special cases of affine transformation. Hence, we can express the geometrical distortion by an affine equation:

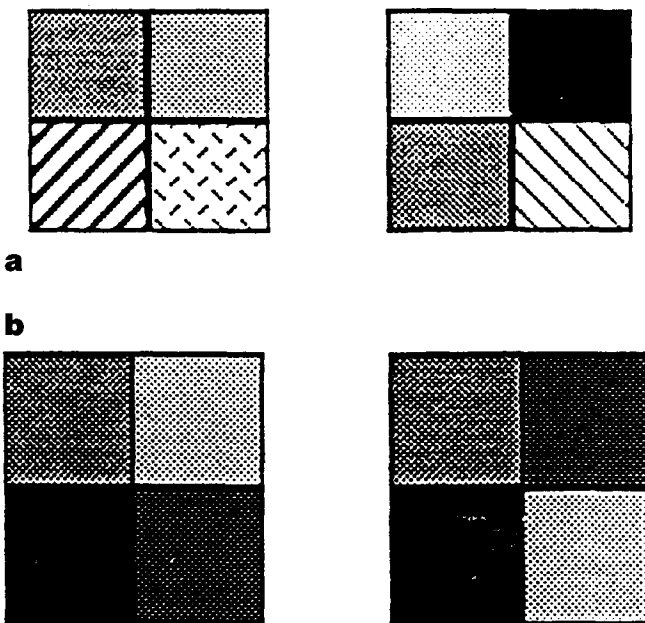


Figure 2 (a) Two patterns with different contents; (b) two patterns with different relative locations among sub-images

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (5)$$

where $[u, v]^T$ denotes the transformed position corresponding to point $[x, y]^T$:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \text{ and } \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \text{ are affine coefficients}$$

Definition The covariance matrix of a given image is defined as:

$$\begin{bmatrix} u_{20} & u_{11} \\ u_{11} & u_{02} \end{bmatrix}$$

and is denoted by \mathbf{M} where u_{ij} is the jointly central moment defined in equation (4).

By the above definition, we derive the following theorem:

Theorem 1 Let \mathbf{M} be the covariance matrix of the original image and \mathbf{M}' be the covariance matrix of the distorted image, which is distorted according to equation (5); then the relationship between \mathbf{M} and \mathbf{M}' is:

$$\mathbf{M}' = \mathbf{A}\mathbf{M}\mathbf{A}^T \quad (6)$$

where:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

Proof By equation (5):

$$u = a_{11}x + a_{12}y + b_1$$

$$v = a_{21}x + a_{22}y + b_2$$

Thus:

$$\begin{aligned} C_u &= E\{u\} \\ &= \int_{\Sigma} u \cdot f_{uv}(u, v) du dv \\ &= \int_{\Omega} (a_{11}x + a_{12}y + b_1) \cdot f_{xy}(x, y) dx dy \\ &= E\{a_{11}x + a_{12}y + b_1\} \\ &= a_{11}C_x + a_{12}C_y + b_1 \end{aligned}$$

where $f_{uv}(u, v)$ is the p.d.f of the distorted image and Σ denotes the region we consider in the distorted image.

Similarly:

$$C_v = a_{21}C_x + a_{22}C_y + b_2$$

Let u'_{ij} denote the covariance of the distorted image; then:

$$\begin{aligned} u'_{20} &= E\{(u - C_u)^2\} \\ &= E\{(a_{11}x + a_{12}y + b_1 - a_{11}C_x - a_{12}C_y - b_1)^2\} \\ &= E\{[a_{11}(x - C_x) + a_{12}(y - C_y)]^2\} \\ &= a_{11}^2 u_{20} + a_{12}^2 u_{02} + 2a_{11}a_{12}u_{11} \end{aligned}$$

Similarly:

$$u'_{11} = a_{11}a_{21}u_{20} + a_{12}a_{22}u_{02} + (a_{11}a_{22} + a_{12}a_{21})u_{11}$$

$$u'_{02} = a_{21}^2 u_{20} + a_{22}^2 u_{02} + 2a_{21}a_{22}u_{11}$$

Combine these variances into vector form:

$$\begin{aligned} \begin{bmatrix} u'_{20} \\ u'_{11} \\ u'_{11} \\ u'_{02} \end{bmatrix} &= \begin{bmatrix} a_{11}^2 & a_{11}a_{12} & a_{11}a_{12} & a_{12}^2 \\ a_{11}a_{21} & a_{11}a_{22} & a_{12}a_{21} & a_{12}a_{22} \\ a_{11}a_{21} & a_{21}a_{12} & a_{11}a_{22} & a_{12}a_{22} \\ a_{21}^2 & a_{21}a_{22} & a_{21}a_{22} & a_{22}^2 \end{bmatrix} \begin{bmatrix} u_{20} \\ u_{11} \\ u_{11} \\ u_{02} \end{bmatrix} \\ &= \begin{bmatrix} a_{11} & a_{11} & a_{12} \\ a_{21} & a_{21} & a_{22} \end{bmatrix} a_{12} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} u_{20} \\ u_{11} \\ u_{11} \\ u_{02} \end{bmatrix} \end{aligned} \quad (7)$$

For kronecker products, equation (7) may be rewritten in the form:

$$\begin{bmatrix} u'_{20} & u'_{11} \\ u'_{11} & u'_{02} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} u_{20} & u_{11} \\ u_{11} & u_{02} \end{bmatrix} \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix} \quad (8)$$

Consequently, $\mathbf{M}' = \mathbf{A}\mathbf{M}\mathbf{A}^T$. We have proved Theorem 1.

ALGORITHM FOR COMPACTING AN IMAGE

In this section, we review the method proposed by Leu⁹ which describes how to compact an image. This procedure has three steps:

1. Computing the covariance matrix \mathbf{M} of a given pattern.
2. Aligning the coordinates with the eigenvectors of \mathbf{M} .
3. Rescaling the coordinates according to the eigenvalues of \mathbf{M} .

Compacted by the algorithm, the pattern is invariant to translation, scaling and skew. Only rotational problem remains unsolved, which we prove in the next section.

We discuss the above three steps in detail in the following subsections.

Computing the covariance matrix of an image

The goal of the algorithm is to adjust an image through a sequence of two linear transformations so that the covariance matrix of the compacted image becomes a scaled identical matrix.

By equations (3) and (4), we can calculate the covariance matrix:

$$\mathbf{M} = \begin{bmatrix} u_{20} & u_{11} \\ u_{11} & u_{02} \end{bmatrix} \quad (9)$$

In pattern recognition, we use the covariance matrix to decouple correlated features and to scale the features to make the clusters compact.

Aligning the coordinates with eigenvectors of \mathbf{M}

In this step, we first find the eigenvalues and eigenvectors of \mathbf{M} , then rotate the coordinate parallel to the eigenvectors of \mathbf{M} .

Assume λ_1 and λ_2 are the eigenvalues of M ; they can be found by solving the equation:

$$\begin{vmatrix} u_{20} - \lambda & u_{11} \\ u_{11} & u_{02} - \lambda \end{vmatrix} = 0 \tag{10}$$

The two roots of the above equation are:

$$\begin{aligned} \lambda_1 &= \frac{u_{20} + u_{02} + \sqrt{(u_{20} - u_{02})^2 + 4u_{11}^2}}{2} \\ \lambda_2 &= \frac{u_{20} + u_{02} - \sqrt{(u_{20} - u_{02})^2 + 4u_{11}^2}}{2} \end{aligned} \tag{11}$$

Hence the eigenvectors corresponding to each eigenvalue can be determined.

Let $e_1 = [e_{1x} e_{1y}]^T$ be the eigenvector associated with λ_1 , and $e_2 = [e_{2x} e_{2y}]^T$ the eigenvector associated with λ_2 . These eigenvectors satisfy the following equations:

$$\begin{vmatrix} u_{20} - \lambda_i & u_{11} \\ u_{11} & u_{02} - \lambda_i \end{vmatrix} \begin{bmatrix} e_{ix} \\ e_{iy} \end{bmatrix} = 0 \text{ for } i = 1, 2 \tag{12}$$

and setting the norm equal to 1:

$$e_{ix}^2 + e_{iy}^2 = 1 \text{ for } i = 1, 2 \tag{13}$$

solving equations (12) and (13), we get:

$$e_i = \begin{bmatrix} e_{ix} \\ e_{iy} \end{bmatrix} = \begin{bmatrix} \frac{u_{11}}{\sqrt{(\lambda_i - u_{20})^2 + u_{11}^2}} \\ \frac{\lambda_i - u_{20}}{\sqrt{(\lambda_i - u_{20})^2 + u_{11}^2}} \end{bmatrix} \tag{14}$$

Consequently, we can construct a rotational matrix E :

$$E = \begin{bmatrix} e_{1x} & e_{1y} \\ e_{2x} & e_{2y} \end{bmatrix} \tag{15}$$

In the procedure, we transform the coordinate system by first translating the origin to the image centre and then multiplying the coordinates with matrix E . Thus, the new coordinates are lying on the eigenvectors of M . Let $[x' y']^T$ denote the new coordinates, then:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = E \begin{bmatrix} x - C_x \\ y - C_y \end{bmatrix} \tag{16}$$

Since the matrix M is real and symmetric, both eigenvectors are orthonormal to each other. Hence, the inner product of the two eigenvectors is zero:

$$e_{1x}e_{1y} + e_{2x}e_{2y} = 0 \tag{17}$$

There are two solutions for the above equation:

$$e_{2x} = e_{1y} \tag{18}$$

$$e_{2y} = -e_{1x}$$

or:

$$e_{2x} = -e_{1y} \tag{19}$$

$$e_{2y} = e_{1x}$$

If the $[e_{2x} e_{2y}]^T$ found in equation (14) is the case of equation (18), the transformed pattern will be reflective to the original pattern. To avoid the reflection, E must be rewritten in the form:

$$E = \begin{bmatrix} e_{1x} & e_{1y} \\ -e_{1y} & e_{1x} \end{bmatrix} \tag{20}$$

By Theorem 1 and equation (16), the covariance matrix of transformed pattern becomes:

$$M' = EME^T = \Lambda \tag{21}$$

where:

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \text{ is a diagonal matrix}$$

Consequently, the pattern becomes uncorrelated to the new coordinate system.

Rescale the coordinates using eigenvalues of M

In the previous step, we rotated the coordinate system so that the image becomes uncorrelated in the new coordinates. Hence, we can scale each component independently. The objective is to get an image whose covariance matrix is equal to a scaled identical matrix. So, we modify the scales of the two new coordinates according to the corresponding eigenvalues:

$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = W \begin{bmatrix} x' \\ y' \end{bmatrix} \tag{22}$$

where:

$$W = \begin{bmatrix} \frac{c}{\sqrt{\lambda_1}} & 0 \\ 0 & \frac{c}{\sqrt{\lambda_2}} \end{bmatrix}, \quad \frac{c}{\sqrt{\lambda_1}} \cdot \frac{c}{\sqrt{\lambda_2}} = 1, c = (\lambda_1, \lambda_2)^{1/4} \tag{23}$$

In this paper, we call W a *scaling matrix*, which preserves the overall size of the normalized shape.

In the above transformation, the covariance matrix becomes:

$$M'' = WM'W^T = c^2 I$$

where I is an identical matrix

Combining equations (16) and (22):

$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = WE \begin{bmatrix} x - C_x \\ y - C_y \end{bmatrix} \tag{24}$$

Thus, we get the most compact image after the translation in equation (24).

Experimental results

1. Curve

Figure 3a is the original curve and Figure 3b is the compact form of Figure 3a by the transformation of equation (24) with $c = 100$.

2. Shape

Figure 4a is the original shape and Figure 4b is its compact form with $c = 30$.

3. Grey-level image

Figure 5a is the original image and Figure 5b is its compact form with $c = 30$.

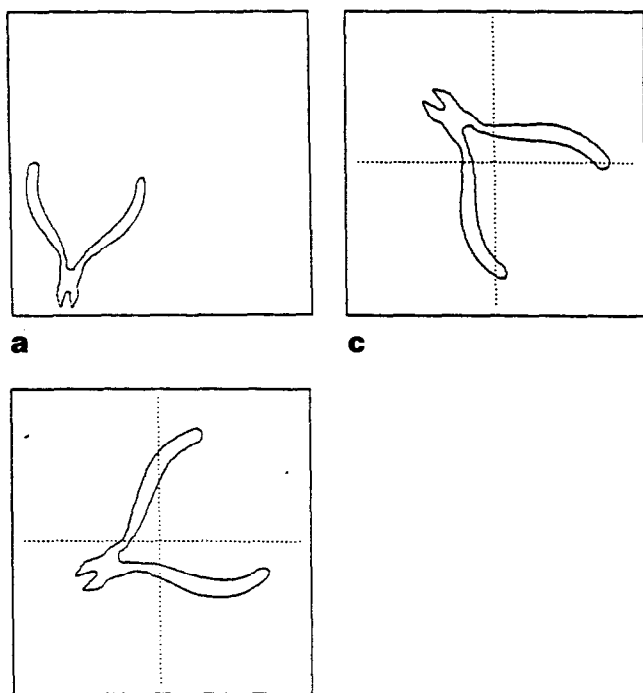


Figure 3 (a) Machine tool; (b) compact curve of Figure 3a; (c) normalized curve of Figure 3a

Note that all the original images are not in their most compact form. Thus, the compact procedure will deform and normalize the original images.

PROPERTIES OF THE COMPACT IMAGE

In general, there are five basic forms of shape distortion caused by the change in the viewer's location: translation, rotation, scaling, skew and reflection. All of these can be represented by the affine transformation:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{25}$$

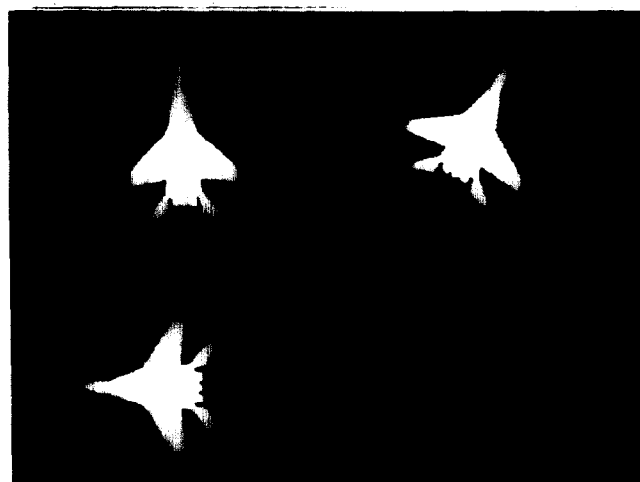


Figure 4 (a) Jet; (b) compact image of (a); (c) normalized image of (a)

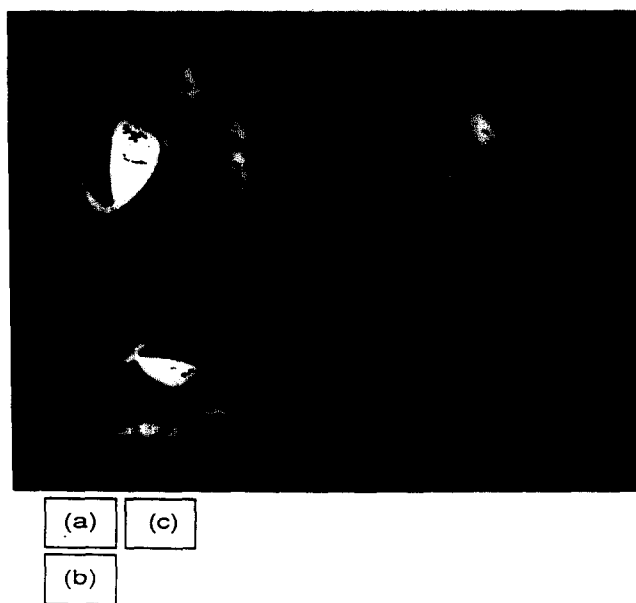


Figure 5 (a) Wai-wai; (b) compact image of (a); (c) normalized image of (a)

or expressed in vector form:

$$\mathbf{u} = \mathbf{Ax} + \mathbf{b} \tag{26}$$

The purpose of this paper is to find a normalization algorithm that can make patterns invariant under translation, rotation, scaling and skew. We do not consider the problem of reflection.

The purpose of the image normalization procedure is to redescribe an image such that the new descriptor is invariant under translation, rotation, scaling and skew. Then, we may apply the descriptor for discrimination.

In the following, we discuss how these distortions affect the compact image.

Translation

Since the compact algorithm translates the origin of the coordinate system to the image centre (see equation (16)), we can easily prove that the method is invariant to translation.

Experimental results

Figure 6a is the translation of Figure 3a and Figure 6b is its compact form. Compared with Figure 3b, the two compact curves are the same, thus the algorithm is invariant under translation.

Scaling

Since the compact algorithm rescales the image to its most compact form with a constant scale, (see equation (22)), it is easy to prove that the method is invariant to scaling.

Experimental results

1. Curve

Figure 7a is the scaling of Figure 3a and Figure 7b is its

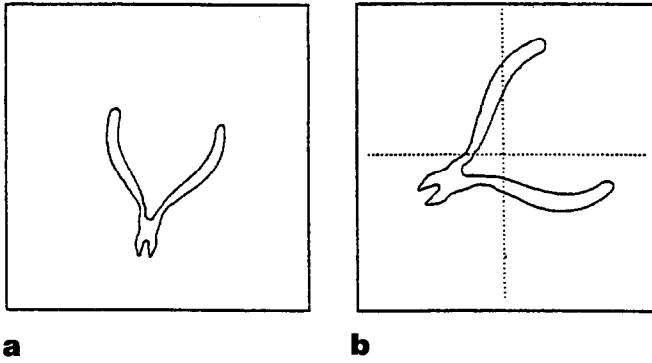


Figure 6 (a) Translation of Figure 3a; (b) compact curve of Figure 6a

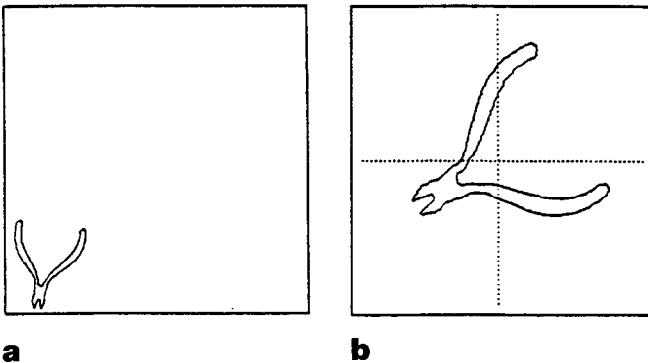


Figure 7 (a) Scaling of Figure 3a; (b) compact curve of Figure 7a

compact form. Compared with Figure 3b, the two compact curves are the same.

2. Shape

Figure 8a is the translation and scaling of Figure 4a, and Figure 8b is its compact form, which is the same as Figure 4b. Thus, the algorithm is invariant under scaling.

Rotation

If a pattern is rotated clockwise by an angle θ , the relation between the rotated and original images is:

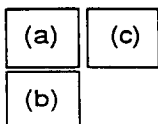
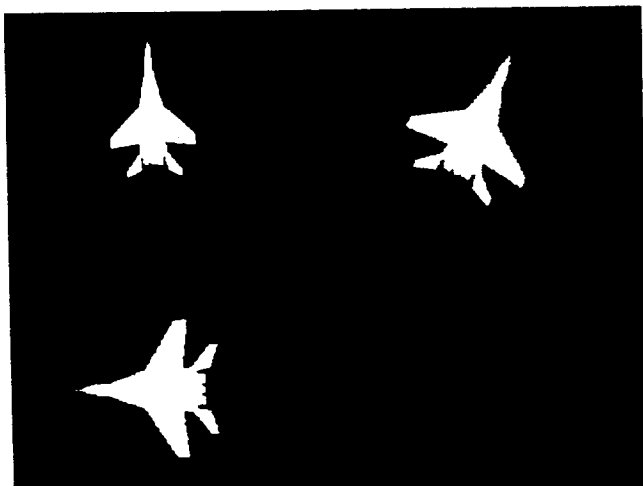


Figure 8 (a) Scaling and shifting of Figure 4a; (b) compact image of (a); (c) normalized image of (a)

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{27}$$

which in vector form is:

$$\mathbf{u} = \mathbf{A}\mathbf{x} \tag{28}$$

where:

$$\mathbf{A} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \text{ is an orthogonal matrix}$$

Since \mathbf{A} is an orthogonal matrix, and by Theorem 1, the relationship between their covariance matrices is:

$$\mathbf{M}_u = \mathbf{A}\mathbf{M}_x\mathbf{A}^T = \mathbf{A}\mathbf{M}_x\mathbf{A}^{-1} \tag{29}$$

Consequently, \mathbf{M}_u is orthogonally similar to \mathbf{M}_x . Both \mathbf{M}_u and \mathbf{M}_x have the same eigenvalues, but their eigenvectors are not equal.

Let \mathbf{E}_u and \mathbf{E}_x denote the rotational matrix corresponding to \mathbf{M}_u and \mathbf{M}_x , respectively. For the decomposition of eigenvalues and eigenvectors (see equations (19) and (21)):

$$\begin{aligned} \mathbf{E}_x\mathbf{M}_x\mathbf{E}_x^T &= \Lambda_x = \Lambda_u \\ &= \mathbf{E}_u\mathbf{M}_u\mathbf{E}_u^T \\ &= \mathbf{E}_u\mathbf{A}\mathbf{M}_x\mathbf{A}^T\mathbf{E}_u^T \\ &= (\mathbf{E}_u\mathbf{A})\mathbf{M}_x(\mathbf{E}_u\mathbf{A})^T \end{aligned} \tag{30}$$

Consequently, we get:

$$\mathbf{E}_u = \mathbf{E}_x\mathbf{A}^{-1} \text{ or } \mathbf{E}_u = -\mathbf{E}_x\mathbf{A}^{-1} \tag{31}$$

The meaning of the above equation is that when the pattern is rotated clockwise by an angle θ , the eigenvectors of the covariance matrix are also rotated by an angle θ or $\theta + \pi$. Hence, we can rotate the pattern back to the original location, or over π by aligning the coordinates on the eigenvectors of the covariance matrix. In the compact algorithm, according to equation (26), the location of the rotated image in compact form is:

$$\begin{aligned} \mathbf{u}'' &= \mathbf{W}_u\mathbf{E}_u(\mathbf{u} - \mathbf{c}_u) \\ &= \mathbf{W}_x(\pm\mathbf{E}_x\mathbf{A}^{-1})(\mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{c}_x) \\ &= \pm\mathbf{W}_x\mathbf{E}_x(\mathbf{x} - \mathbf{c}_x) \\ &= \pm\mathbf{x}'' \end{aligned} \tag{32}$$

Thus, the compact algorithm makes the rotated image turn back to the original location, or over π .

Experimental results

1. Curves

Figure 9a is the rotation of the Figure 3a and Figure 9b is its compact form which is the same as Figure 3b. Figure 10a is the rotation of Figure 3a and Figure 10b is its compact form, which is differentiated from Figure 3b by the angle π .

2. Shapes

Figure 11a is the rotation of Figure 4a and Figure 11b is its compact form, which is the same as Figure 4b.

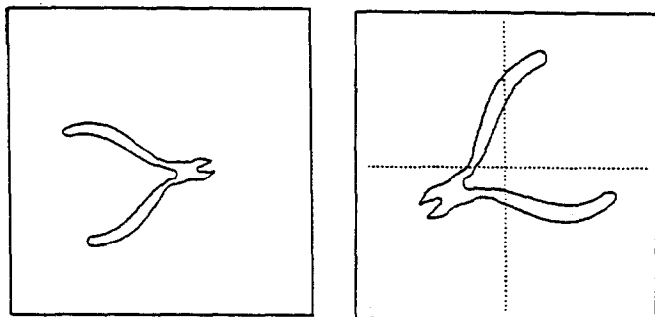


Figure 9 (a) Rotation of Figure 3a; (b) compact curve of Figure 9a

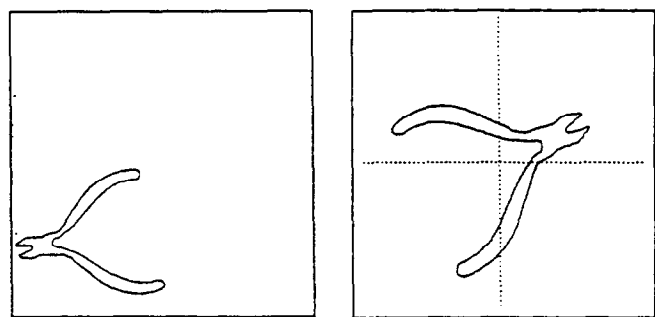


Figure 10 (a) Rotation of Figure 3a; (b) compact curve of Figure 10a

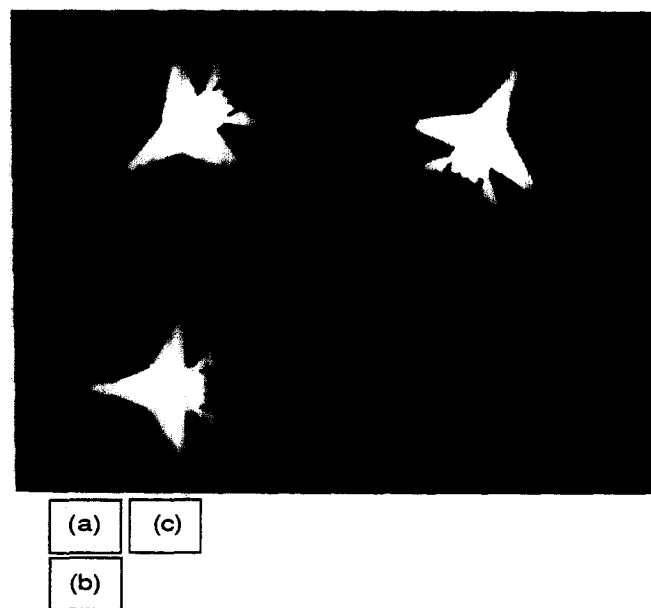


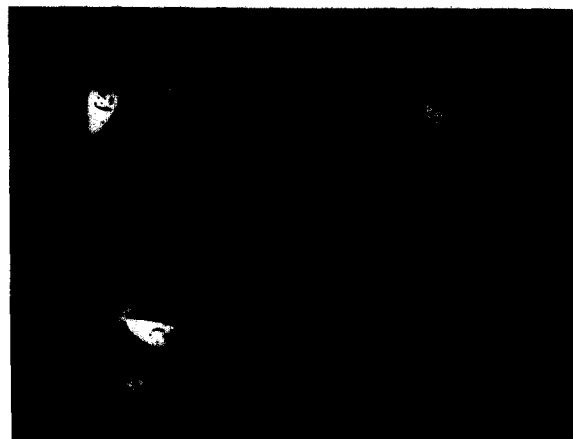
Figure 11 (a) Rotation of Figure 4a; (b) compact image of (a); (c) normalized image of (a)

3. Grey-level images

Figure 12a is the translation, rotation and scaling of Figure 5a and Figure 12b is its compact form, which is the same as Figure 5b.

Figure 13a is the rotation of Figure 5a and Figure 13b is its compact form, which is differentiated from Figure 5b by the angle π .

Thus, the compact algorithm makes the rotated images turn back the same location, or over π .



(a) (c)
(b)

Figure 12 (a) Shifting, rotation and scaling of Figure 5a; (b) compact image of (a); (c) normalized image of (a)



(a) (c)
(b)

Figure 13 (a) Rotation of Figure 5a; (b) compact image of (a); (c) normalized image of (a)

Skew

The model of the skew transformation is:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (33)$$

or in vector form:

$$u = Ax = BCx \quad (34)$$

where:

$$B = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \text{ and } a \neq b \quad C = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}$$

$$A = BC$$

The matrix C determines in which direction the image is

skewed, and the matrix B determines how severely the image is skewed.

Since A is a general matrix, the rotational and scaling matrices of the skewed and original images are not equal, respectively. By equation (21), we get:

$$M = E^T \Lambda E \tag{35}$$

and by equation (23), the relation between the scaling matrix and the eigenvalues of M is:

$$W = c \Lambda^{-1/2} \tag{36}$$

The location of the skewed image in compact form is:

$$u'' = W_u E(u - c_u)$$

Combined with equation (36), we get:

$$u'' = \Lambda_u^{-1/2} E_u A E_x^T \Lambda_x^{1/2} x'' \tag{37}$$

Define $G = \Lambda_u^{-1/2} E_u A E_x^T \Lambda_x^{1/2}$; combined with equation (35), we can prove:

$$G G^T = I \tag{38}$$

Consequently, G is an orthogonal matrix.

We know that any two-dimensional orthogonal matrix may be written in the form:

$$\begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \tag{39}$$

or:

$$\begin{bmatrix} \cos \alpha & \sin \alpha \\ \sin \alpha & -\cos \alpha \end{bmatrix} \tag{40}$$

We confine the rotational matrix to be equal to equation (20), to avoid the reflection of the pattern. Thus, G is of the form of equation (17), not as equation (40). Hence, the compact algorithm turns the skew problem into a rotational problem.

Experimental results

1. Curve

Figure 14a is the skew of Figure 3a and Figure 14b is its compact form, which is different from Figure 3b by rotation.

2. Shape

Figure 15a is the skew of Figure 4a and Figure 15b is its compact form, which is different from Figure 4b by rotation only.

3. Grey-level image

Figure 16a is the skew of Figure 5a and Figure 16a is its compact form, which is different from figure 5b by rotation only. Thus, the compact algorithm turns the skew problem into a rotational problem.

FINAL STEP FOR IMAGE NORMALIZATION

Concluding the discussion in the preceding sections, the pattern is invariant under translation, scaling and skew. After the compacting process, only the rotational problem remains unsolved. In this section, we propose two methods to solve this problem.

Method 1: Find the furthest point

We may easily determine the point that is furthest from the centre of mass of the shape. As shown in Figure 17, let point A be the furthest point in the shape. We can measure the angle between the directions of OA and the x-axis, then rotate the shape clockwise by this angle. We call the rotated compact image a *normalized* image, which is invariant under translation, rotation, scaling and skew.

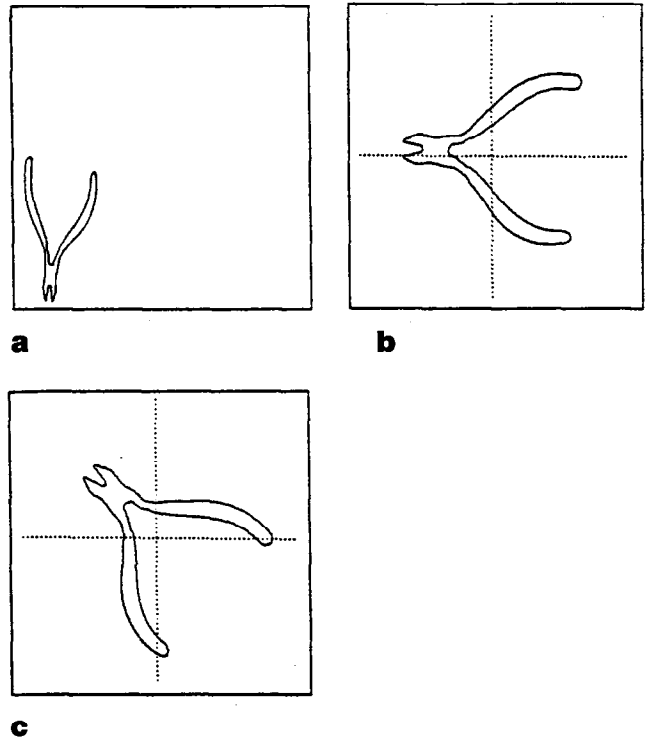


Figure 14 (a) Skew of Figure 3a; (b) compact of Figure 14a; (c) normalized curve of Figure 14a

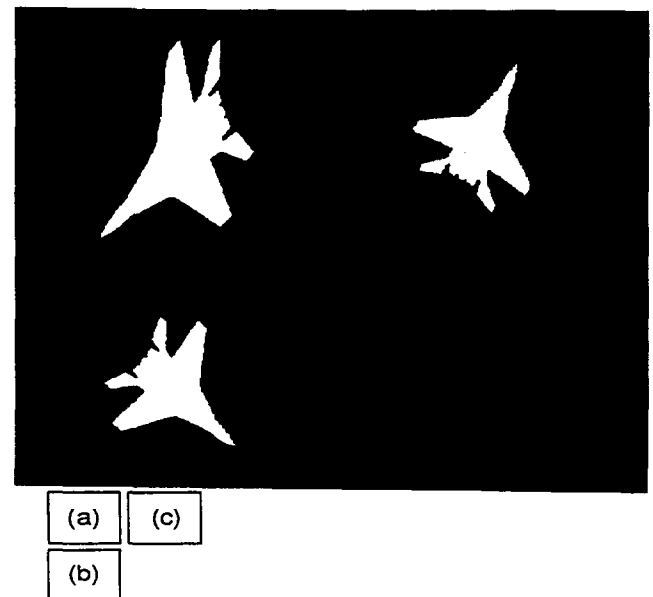


Figure 15 (a) Skew of Figure 4a; (b) compact image of (a); (c) normalized image of (a)

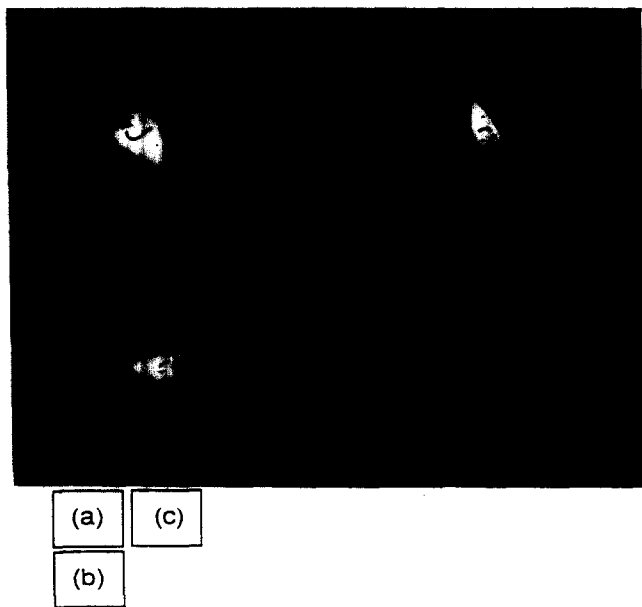


Figure 16 (a) Skew of Figure 5a; (b) compact image of (a); (c) normalized image of (a)

Experimental results

1. Curve

Figures 3c, 14c and 18b are the normalized curves of Figures 3a, 14a and 18a, respectively. Observe that all the normalized curves are the same, hence, the normalized curves are invariant under translation, rotation, scaling and skew.

Although this method is simple, it fails when the shape is noisy or the furthest point is not unique. Another defect is that this method is only suitable for curve or shape. When the image is a grey-level or coloured image, it becomes difficult to determine the object's region; consequently, it is difficult to determine which is the furthest point.

Method 2: Normalization by tensor theory

As the method makes use of some tensor theories, we present the tensor representation of an image and some theories¹⁰⁻¹² and apply the tensor theories to normalized images below.

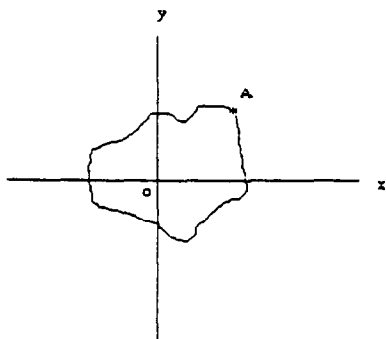


Figure 17 Find the furthest point A for rotation normalization

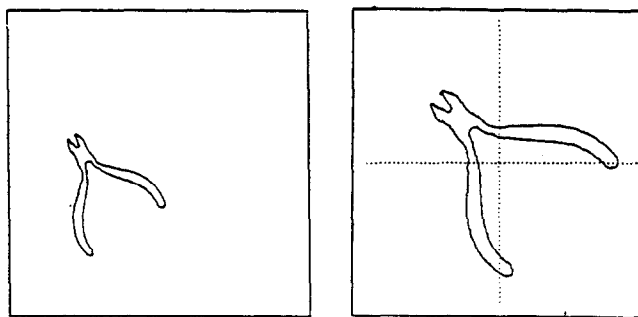


Figure 18 (a) Translation, rotation and scaling of Figure 3a; (b) normalized curve of Figure 18a

Formulation of moment tensors

In the above sections, we express the affine transformation as:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{41}$$

the first order moments of the image are denoted by $[C_x \ C_y]^T$ and the jointly central moments are denoted by u_{kr} . If we translate the coordinates to the centre of the image, equation (41) becomes:

$$\begin{bmatrix} u - C_u \\ v - C_v \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x - C_x \\ y - C_y \end{bmatrix} \tag{42}$$

Apparently, the above equation is invariant to translation.

In the following discussion, the original point of coordinates is assumed to be at the centre of the image to simplify the notation.

In tensor notation, the coordinate variables are different by an index, so that x^1 here is equivalent to the x -coordinate, and x^2 is equivalent to the y -coordinate. Thus, the above equation becomes:

$$\begin{bmatrix} y^1 \\ y^2 \end{bmatrix} = \begin{bmatrix} A_1^1 & A_2^1 \\ A_1^2 & A_2^2 \end{bmatrix} \begin{bmatrix} x^1 \\ x^2 \end{bmatrix} \tag{43}$$

where:

$$\begin{bmatrix} y^1 \\ y^2 \end{bmatrix} = \begin{bmatrix} u - C_u \\ v - C_v \end{bmatrix} \text{ and } \begin{bmatrix} x^1 \\ x^2 \end{bmatrix} = \begin{bmatrix} x - C_x \\ y - C_y \end{bmatrix}$$

$$\begin{bmatrix} A_1^1 & A_2^1 \\ A_1^2 & A_2^2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

In a tensor expression, equation (43) is denoted by:

$$y_i = A_j^i x^j \text{ for } i = 1, 2 \tag{44}$$

Hence, we may define the inverse affine transformation:

$$x_i = a_j^i y^j \text{ for } i = 1, 2 \tag{45}$$

with the property:

$$A_j^i a_k^j = \begin{cases} 1 & \text{if } i = k \\ 0 & \text{if } i \neq k \end{cases} = \delta_{ik} \tag{46}$$

The central moment tensors are defined as:

$$T^{ijk\dots} = \int_{\Omega} x^i x^j x^k \dots f(x^1, x^2) dx^1 dx^2 \quad (47)$$

This type of tensor is the case of contravariant tensor operators. We derive a theorem:

Theorem 2 *Let:*

$$T^{ijk\dots} = \int_{\Omega} x^i x^j x^k \dots f(x^1, x^2) dx^1 dx^2$$

$$\bar{T}^{ijk\dots} = \int_{\Sigma} y^i y^j y^k \dots f(y^1, y^2) dy^1 dy^2$$

if $(x^i x^j x^k)$ and $(y^i y^j y^k)$ are related as tensors, i.e.:

$$y^i y^j y^k \dots = A_i^l A_j^m A_k^n \dots x^l x^m x^n \dots$$

then $(T^i T^j T^k)$ and $(\bar{T}^i \bar{T}^j \bar{T}^k)$ are also related as tensors, i.e.:

$$\bar{T}^i \bar{T}^j \bar{T}^k \dots = A_i^l A_j^m A_k^n \dots T^l T^m T^n \dots$$

Another type of tensor is the case of covariant tensor operators; we denote the index in the subscript, i.e.:

$$T_{ijk\dots} = \int_{\Omega} x^i x^j x^k \dots f(x^1, x^2) dx^1 dx^2 \quad (48)$$

$$\bar{T}_{ijk\dots} = \int_{\Sigma} y^i y^j y^k \dots f(y^1, y^2) dy^1 dy^2 \quad (49)$$

It also has a theorem:

Theorem 3 *If $(y_i y_j y_k \dots)$ and $(x_i x_j x_k \dots)$ are related as tensors, i.e.:*

$$x_i x_j x_k \dots = a_i^l a_j^m a_k^n \dots y_l y_m y_n \dots$$

then $(\bar{T}_i \bar{T}_j \bar{T}_k \dots)$ and $(T_i T_j T_k \dots)$ are also related as tensors, i.e.:

$$T_i T_j T_k \dots = a_i^l a_j^m a_k^n \dots \bar{T}_l \bar{T}_m \bar{T}_n \dots$$

Another important feature in tensors is the permutation tensor, which is defined as ε_{ij} with:

$$\begin{cases} \varepsilon_{11} = 0 \\ \varepsilon_{12} = 1 \\ \varepsilon_{21} = -1 \\ \varepsilon_{22} = 0 \end{cases} \quad (50)$$

It has the property:

$$\bar{\varepsilon}_{ij} = J a_i^l a_j^m \varepsilon_{lm} \quad (51)$$

where J is the Jacobian of the affine transformation:

$$J = \begin{vmatrix} A_1^1 & A_2^1 \\ A_1^2 & A_2^2 \end{vmatrix} = \begin{vmatrix} a_1^1 & a_2^1 \\ a_1^2 & a_2^2 \end{vmatrix}^{-1} \quad (52)$$

For the case $J = 1$, $\bar{\varepsilon}_{ij} = a_i^l a_j^m \varepsilon_{lm}$. Hence ε_{ij} and $\bar{\varepsilon}_{ij}$ are related to tensors

By combing the permutation tensors, we may reduce the higher order tensors to lower order ones. For example, we define a tensor:

$$t^m = T^{ij} \varepsilon_{ik} \varepsilon_{jl} T^{klm} / \Delta^2 \quad (53)$$

where Δ is the area of an object's image.

Let $\bar{t}^m = \bar{T}^{ij} \bar{\varepsilon}_{ik} \bar{\varepsilon}_{jl} \bar{T}^{klm} / \bar{\Delta}^2$. By Theorems 2 and 3, we can derive that:

$$\bar{t}^m = A_k^m t^k \quad (54)$$

Thus, we reduce the second and third order tensors to a first order tensor; a linear equation.

Normalization of a compact image

Since the only problem remaining is rotation, we just find an angle by which we may rotate the compact image into a normalized range which is invariant under rotation. The diagram of this is shown in Figure 19.

In tensor notation:

$$\begin{bmatrix} A_1^1 & A_2^1 \\ A_1^2 & A_2^2 \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix}$$

Hence the Jacobian $J = 1$.

Since the Jacobian $J = 1$, $\bar{\varepsilon}_{ij}$ and ε_{ij} are related to tensors. We define a first-order tensor:

$$t^m = T^{ij} \varepsilon_{ik} \varepsilon_{jl} T^{klm} \quad (55)$$

By Theorem 2, it has the property:

$$\bar{t}^m = A_k^m t^k \quad (56)$$

Since the covariance matrix of a compact image becomes a scaled identical matrix $c^2 I$, the second order tensor in equation (47) becomes:

$$T^{12} = T^{21} = 0 \quad (57)$$

and:

$$T^{11} = T^{22} = c^2 \quad (58)$$

Substituting equations (57) and (58) in equation (56), we get:

$$t^m = c^2 (T^{22m} + T^{11m}) \quad (59)$$

Rewriting equation (57) by the central moments defined earlier we get:

$$t^1 = c^2 (T^{221} + T^{111}) = c^2 (u_{12} + u_{30}) \quad (60)$$

$$t^2 = c^2 (T^{222} + T^{112}) = c^2 (u_{03} + u_{21}) \quad (61)$$

Note that u_{ij} is the central moment of the compact image.

Since t^m is the first-order tensor, equation (56) is equivalent to:

$$\begin{bmatrix} \bar{t}^1 \\ \bar{t}^2 \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} t^1 \\ t^2 \end{bmatrix} \quad (62)$$

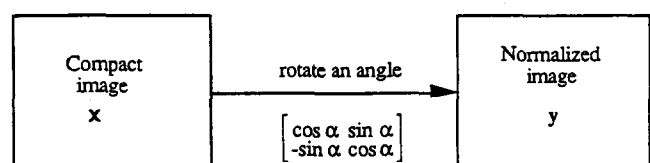


Figure 19 Find an angle α for rotation normalization

where \bar{t}^1 and \bar{t}^2 are the tensors of the normalized image, and t^1 and t^2 are the tensors of the compact image.

For the normalization algorithm, we rotate the compact image around the centre point of the image such that the tensor \bar{t}^1 becomes zero. Thus, by setting $\bar{t}^1 = 0$, we get:

$$t^1 = 0 = t^1 \cos \alpha + t^2 \sin \alpha \quad (63)$$

Substituting equations (60) and (61) into equation (58), we get the relation:

$$\tan \alpha = -\frac{t^1}{t^2} = -\frac{u_{12} + u_{30}}{u_{03} + u_{21}} \quad (64)$$

Apparently, there are two solutions to the above equation, say ϕ and $\phi + \pi$.

Substituting ϕ or $\phi + \pi$ into equation (62), we get:

$$\bar{t}^2(\phi) = -\bar{t}^2(\phi + \pi) \quad (65)$$

The meaning of the above equation is that one of the solutions of equation (64) makes $\bar{t}^2 > 0$, and the other makes $\bar{t}^2 < 0$. Thus, in the algorithm we choose ϕ_0 in $\{\phi, \phi + \pi\}$ such that $\bar{t}^2 > 0$. By this option, we may uniquely determine the rotational angle.

Combining with equation (24), we get the transformation equation from the original to the normalized image:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \frac{c}{\sqrt{\lambda_1}} & 0 \\ 0 & \frac{c}{\sqrt{\lambda_2}} \end{bmatrix} \begin{bmatrix} e_{1x} & e_{1y} \\ -e_{1y} & e_{1x} \end{bmatrix} \begin{bmatrix} x - C_x \\ y - C_y \end{bmatrix} \quad (66)$$

By the transformation, we can get the normalized image which is invariant under translation, rotation, scaling and skew. The normalized procedures for affine-distorted pattern recognition are consistent with the mental transformation theory in psychology and psychophysics. Shepard and Cooper¹³ provide empirical evidence in support of the mental transform theory in human perception.

SUMMARY OF NORMALIZATION ALGORITHM

In this section, we summarize the normalization procedure. The steps are:

1. Computing the mean vector c and the covariance matrix M of the original image:

$$c = [C_x \ C_y]^T \quad (67)$$

and:

$$M = \begin{bmatrix} u_{20} & u_{11} \\ u_{11} & u_{02} \end{bmatrix} \quad (68)$$

2. Aligning the coordinates with the eigenvectors of M :

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} e_{1x} & e_{1y} \\ -e_{1y} & e_{1x} \end{bmatrix} \begin{bmatrix} x - C_x \\ y - C_y \end{bmatrix} \quad (69)$$

3. Rescaling the coordinates according to the eigenvalues of M :

$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} \frac{c}{\sqrt{\lambda_1}} & 0 \\ 0 & \frac{c}{\sqrt{\lambda_2}} \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} \quad (70)$$

To save computation time, we may combine step (2) with step (3):

$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} \frac{c}{\sqrt{\lambda_1}} & 0 \\ 0 & \frac{c}{\sqrt{\lambda_2}} \end{bmatrix} \begin{bmatrix} e_{1x} & e_{1y} \\ -e_{1y} & e_{1x} \end{bmatrix} \begin{bmatrix} x - C_x \\ y - C_y \end{bmatrix} \quad (71)$$

We call the image transformed by equation (71) a *compact image*.

4. Computing the third-order central moments of the compact image, say:

$$u_{30}, u_{21}, u_{12}, u_{03} \quad (72)$$

5. Calculating the tensors t^1 and t^2 :

$$t^1 = u_{12} + u_{30} \quad (73)$$

$$t^2 = u_{03} + u_{21} \quad (74)$$

6. Finding the angle α , which satisfies the equation:

$$\tan \alpha = -\frac{t^1}{t^2} \quad (75)$$

7. Calculating the tensor \bar{t}^2 :

$$\bar{t}^2 = -t^1 \sin \alpha + t^2 \cos \alpha \quad (76)$$

if $\bar{t}^2 < 0$, then $\alpha = \alpha + \pi$.

8. Rotating the compact image clockwise by α angle:

$$\begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} t^1 \\ t^2 \end{bmatrix} \begin{bmatrix} x'' \\ y'' \end{bmatrix} \quad (77)$$

Thus, we get the normalized image which is invariant to translation, rotation, scaling and skew.

Since the data of the image is large, the fetching and storage of image data in memory takes much of the computing time. By combining steps (2), (3) and (4), we may compute the third-order moments of the compact image directly from the original image. Consequently, we may save computing time in the fetching and storage of the compact image. Another advantage is that, since the location and content of the image is stored in a digital way, we may avoid the quantization error of the moments which is caused by the data of compact image. The details of combining steps (2), (3) and (4) are as follows.

Consider equation (71), which may be expanded as:

$$\begin{aligned} \begin{bmatrix} x'' \\ y'' \end{bmatrix} &= \begin{bmatrix} \frac{c}{\sqrt{\lambda_1}} & 0 \\ 0 & \frac{c}{\sqrt{\lambda_2}} \end{bmatrix} \begin{bmatrix} e_{1x} & e_{1y} \\ -e_{1y} & e_{1x} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \\ &+ \begin{bmatrix} \frac{c}{\sqrt{\lambda_1}} & 0 \\ 0 & \frac{c}{\sqrt{\lambda_2}} \end{bmatrix} \cdot \begin{bmatrix} e_{1x} & e_{1y} \\ -e_{1y} & e_{1x} \end{bmatrix} \cdot \begin{bmatrix} -C_x \\ -C_y \end{bmatrix} \\ &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \end{aligned} \tag{78}$$

where:

$$\begin{aligned} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} &= \begin{bmatrix} \frac{c}{\sqrt{\lambda_1}} & 0 \\ 0 & \frac{c}{\sqrt{\lambda_2}} \end{bmatrix} \begin{bmatrix} e_{1x} & e_{1y} \\ -e_{1y} & e_{1x} \end{bmatrix} \\ &= \begin{bmatrix} \frac{c}{\sqrt{\lambda_1}} e_{1x} & \frac{c}{\sqrt{\lambda_1}} e_{1y} \\ -\frac{c}{\sqrt{\lambda_2}} e_{1y} & \frac{c}{\sqrt{\lambda_2}} e_{1x} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \\ &= \begin{bmatrix} \frac{c}{\sqrt{\lambda_1}} & 0 \\ 0 & \frac{c}{\sqrt{\lambda_2}} \end{bmatrix} \begin{bmatrix} e_{1x} & e_{1y} \\ -e_{1y} & e_{1x} \end{bmatrix} \begin{bmatrix} -C_x \\ -C_y \end{bmatrix} \\ &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} -C_x \\ -C_y \end{bmatrix} \end{aligned}$$

Let $u_{30}u_{21}u_{12}u_{03}$ be the central moments of the original image, and $u'_{30}u'_{21}u'_{12}u'_{03}$ be the central moments of the compact image. By Theorem 1 and equation (78), we get:

$$u'_{30} = a_{11}^3 u_{30} + 3a_{11}^2 a_{12} u_{21} + 3a_{11} a_{12}^2 u_{12} + a_{12}^3 u_{03} \tag{79}$$

$$\begin{aligned} u'_{21} &= a_{11}^2 a_{21} u_{30} + (a_{11}^2 a_{22} + 2a_{11} a_{12} a_{21}) u_{21} \\ &+ (2a_{11} a_{12} a_{22} + a_{12}^2 a_{21}) u_{12} + a_{12}^2 a_{22} u_{03} \end{aligned} \tag{80}$$

$$\begin{aligned} u'_{12} &= a_{11} a_{21}^2 u_{30} + (a_{21}^2 a_{12} + 2a_{11} a_{21} a_{22}) u_{21} \\ &+ (2a_{12} a_{21} a_{22} + a_{22}^2 a_{21}) u_{12} + a_{12} a_{22}^2 u_{03} \end{aligned} \tag{81}$$

$$u'_{03} = a_{21}^3 u_{30} + 3a_{21}^2 a_{22} u_{21} + 3a_{21} a_{22}^2 u_{12} + a_{22}^3 u_{03} \tag{82}$$

Hence, the real algorithm is that:

1. Compute the following moments of an original image
 - (a) mean vector c
 - (b) covariance matrix M
 - (c) third-order central moments $u_{30}u_{21}u_{12}u_{03}$.
2. Calculate the eigenvalues $\lambda_1 \lambda_2$ and eigenvectors $[e_{1x} e_{1y}]^T$ of M .
3. Compute the matrix:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

in equation (78). Then compute the moments of compact image from equation (79) to equation (82).

4. Calculate the tensors t^1 and t^2 :

$$t^1 = u'_{12} + u'_{30}$$

$$t^2 = u'_{03} + u'_{21}$$

5. Find the angle α , which satisfies the equation:

$$\tan \alpha = -\frac{t^1}{t^2}$$

6. Calculate the tensor \bar{t}^2 :

$$\bar{t}^2 = -t^1 \sin \alpha + t^2 \cos \alpha$$

If $\bar{t}^2 < 0$ then $\alpha = \alpha + \pi$.

7. Consequently, the normalized image may be found directly from the original image by the transformation:

$$\begin{aligned} \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} &= \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} \frac{c}{\sqrt{\lambda_1}} & 0 \\ 0 & \frac{c}{\sqrt{\lambda_2}} \end{bmatrix} \\ &\cdot \begin{bmatrix} e_{1x} & e_{1y} \\ -e_{1y} & e_{1x} \end{bmatrix} \cdot \begin{bmatrix} x - C_x \\ y - C_y \end{bmatrix} \end{aligned} \tag{83}$$

Experimental results

1. Shape

Figures 4c, 8c, 11c and 15c are the normalized shapes of Figures 4a, 8a, 11a and 15a respectively. Observe all the normalized shapes, they are the same. Thus the normalized shapes are invariant under translation, rotation, scaling and skew.

2. Grey-level image

Figures 5c, 12c, 13 and 16c are the normalized images of Figures 5a, 12a, 13a and 16a, respectively. Observe that all the normalized images are the same. Thus, the normalized images are invariant under translation, rotation, scaling and skew.

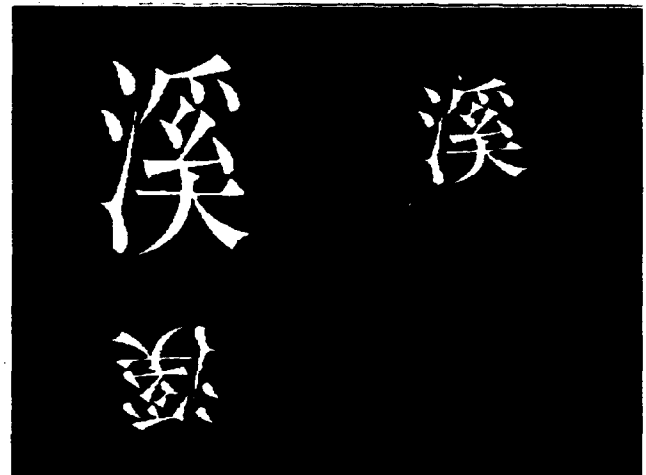


Figure 20 (a) Chinese word 'stream'; (b) compact image of (a); (c) normalized image of (a)

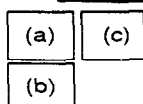


Figure 21 (a) Shifting, rotation and scaling of *Figure 20a*; (b) compact image of (a); (c) normalized image of (a)

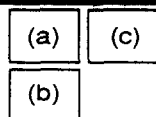


Figure 22 (a) Skew of *Figure 20a*; (b) compact image of (a); (c) normalized image of (a)

3. *Figures 20–22* are other examples. By observing the results, we confirm that this normalization algorithm is very useful and may be applied widely.

The above experiments are simulated on a Microvax model 3600 computer, and the normalized procedures can be done very quickly within 1 min. In real image experiments, one must first extract and isolate the object from the background and place the object for normalization and recognition. For the normalization of colour images, since the RGB components of the original image are not the same, the normalized image of each component is different. Hence, these components cannot match together. One solution is to transform the RGB system into the YIQ system. The Y-component is the luminance of the colour image; we normalize the Y component only, and fix I and Q. The

resulting normalized colour images obtained by experimentation are quite satisfactory.

CONCLUSION

Over the years, a large number of recognition methods have been developed to solve the distortion of translation, rotation and scaling. The method most used has been the statistical method, which first calculates the values of moment invariants¹, then recognizes by statistical classification¹⁴. This method is only suitable when patterns are not large. When patterns are large, the decision rule will become complex and the error of recognition will increase.

The matching method is very easy and suitable for large patterns, but the method must previously normalize the image for matching. In this paper, we have expanded the compact algorithm proposed by Leu⁹ to a more perfect and realizable method which can normalize the image under the distortion of translation, rotation, scaling and skew. There are some advantages of this method:

1. The method is suitable when patterns are large.
2. The normalization algorithm is easy and does need not much computation.
3. The similarity measure by matching is rapid.
4. The searching in the database is efficient.

In summary, image normalization is very useful in image understanding systems.

REFERENCES

- 1 Hu, M K 'Visual pattern recognition by moment invariants', *IRE Trans. Infor. Theory* (February 1962) pp 179–187
- 2 Zahn, C T and Roskies, R S 'Fourier descriptors for plane closed curves', *IEEE Trans. Comput.*, Vol 21 (March 1972) pp 269–281
- 3 Persoon, E and Fu, K S 'Shape discrimination using Fourier descriptors', *IEEE Trans. Syst., Man, Cybern.*, Vol 7 (March 1977) pp 170–179
- 4 Ballard, D H 'Generalizing the Hough transform to detect arbitrary shape', *Patt. Recogn.*, Vol 13 No 2 (1981) pp 111–122
- 5 Taza, A and Suen, C Y 'Discrimination of planar shapes using shape matrices', *IEEE Trans. Syst., Man, Cybern.*, Vol 19 No 5 (September/October 1989) pp 1281–1289
- 6 Rosenfeld, A and Kak, A C *Digital Picture Processing*, Vol 2, Academic Press, New York (1982)
- 7 Cyganski, D and Orr, J A 'Applications of tensor theory to object recognition and orientation determination', *IEEE Trans. PAMI*, Vol 7 No 6 (November 1985) pp 662–673
- 8 Teh, C H and Chin, R T 'On image analysis by the methods of moments', *IEEE Trans. PAMI*, Vol 10 No 4 (1988) pp 496–513
- 9 Leu, J-G 'Shape normalization through compacting', *Patt. Recogn. Lett.*, Vol 10 (October 1989) pp 243–250
- 10 Fang, M and Hansler, G 'Class of transforms invariant under shift, rotation and scaling', *Appl. Optics*, Vol 29 No 5 (February 1990) pp 704–708
- 11 Synge, J L and Schild, A *Tensor Calculus*, Dover, New York (1978)
- 12 Lovelock, D and Rund, H *Tensors, Differential Forms, and Variational Principles*, Wiley, New York (1975)
- 13 Shepard, R N and Cooper, L A *Mental Images and their Transformations*, MIT Press, Cambridge, MA (1986)
- 14 Duda, R O and Hart, P E *Pattern Recognition and Scene Analysis*, John Wiley, New York (1973)