DIRECTION CODING METHOD AND ITS APPLICATION TO SCENE ANALYSIS *

Haruo Yoda, Jun Motoike, Masakazu Ejiri
Central Research Laboratory
Hitachi Ltd.
Kokubunji, Tokyo 185, Japan

Abstract

In order to cope with image signal variations in imaging devices, a direction coding method is proposed for stable scene analysis and object recognition. This method makes extensive use of macroscopic processing of direction-coded images to achieve noise-free analysis. Examples showing recognition of obliquely-imaged box-like objects whose surfaces are rather complicated are included. Additional efforts are made to recognize surface information of objects by spatially transforming the surface image as if it were viewed from a vertical direction. This method also involves position-invariant and rotation-invariant image data transformation techniques, thus raising the possibility of more effective image analysis in actual application.

1. Introduction

Scene analysis techniques in the artificial intelligence field intrinsically have the possibility of being applied to the automation of production lines and physical distribution systems. Many studies have so far been devoted to this area, and have led to the development of prototype intelligent robot systems (1)(2)(3). In the past, we at Hitachi Central Research Laboratory have also made efforts toward developing such robot systems (4) (5) as well as toward realizing industrial eyes involving scene analysis capabilities (6) (7). A number of these have already found practical use in the automation of actual assembly and inspection processes which required unique application of visual technology.

From this experience, we have noted that the variation of image signals from imaging devices is usually the most significant problem to be overcome whenever we attempt to put scene analysis techniques into practical use. This variation is mainly caused by subtle changes in lighting conditions as well as by the intrinsic characteristics of the imaging device itself as affected by fluctuations in power source and temperature. This variation sometimes has ci highly significant influence upon the effectiveness of the scene analysis algorithms, by which objects in the image are ultimately recognized.

In this paper, to begin with an image processing method that can yield stable results all of the time by minimizing the effect of image variation is proposed. Next, an attempt at object recognition by this method is described. The objects handled in this study are mainly parallelepipeds. However, unlike ordinary blocks with blank surfaces, boxes with rather complex pattern information on their surfaces were used. The method proposed here is concerned with how all geometric features, such as position and posture of the object, can be extracted by eliminating the surface information, and how this information on the objects surface can then be recognized.

This method attempted to make maximum use of macroscopic image processing, such as homogeneous processing, frequency distribution calculation, etc., as opposed to microscopic processing such as edge following. Since, it seems that macroscopic image processing is a prerequisite for high speed, stable recognition hardware to be developed in the future.

2_-_Direction Coding Method

2.1 Basis of direction coding

It has been demonstrated that the image signal obtained from an imaging device is affected by a variety of factors. The form of the observable brightness signal g(x,y) is usually expressed as follows:

$$g(x,y) = a\, f(x,y) + b \quad , \qquad (1)$$

where the parameters a and b denote the contrast and level change respectively. In other words, even if the same scene is observed, the same image value is not expected all the time.

Another important characteristic of the image signal is that the value at each picture element in an image does not possess any specific meaning in the geometric sense. Obviously, only the relation of the image values of a picture element and its neighbors can give geometric meaning. This implies that to extract geometric features from an image it is necessary to provide a means to execute comparison with surrounding picture elements.

The two above facts are drawbacks associated with usual image processing when a brightness image itself is used as an input image. They sometimes make the image processing very complicated.

This work is contracted with the Agency of Industrial Science and Technology, Ministry of international Trade and Industry, as at part of the National Research and Development Program "Pattern Information Processing System".

Consequently, the input brightness image is first transformed into what we call a "direction-coded image" on which almost all subsequent processing is based. In this image each picture element has a geometric meaning.

Similar to the methods that are dealt with in previous papers (8)(9), a direction-coded image is one whose value at each picture element corresponds to the tangential direction of brightness change in the original image. That is, the direction-coded image A(x,y) is obtained by performing the following transformation on the original brightness image g(x,y) :

$$\mathbf{V}(x,y) = \left(\frac{\partial g(x,y)}{\partial x}, \frac{\partial g(x,y)}{\partial y} \right)$$

$$A(x,y) = \mathrm{Arg}\left(\mathbf{V}(x,y) \right) + \frac{\pi}{2} \quad , \qquad (2)$$

$$\mathrm{mod.}\ 2\pi .$$

In other words, a direction-coded image is one that has $\frac{\pi}{2}$ added to the argument of the gradient vector of g(x,y). The reason for adding $\frac{\pi}{2}$ is simply because the direction of the border line of the brightness change gives a clearer understanding than the direction of the brightness change itself. As a result, the parameters a and b in eq.(1) are eliminated in eq.(2), thus giving a characteristic image that is not affected in principle by the contrast and black-level change of the imaging device.

In this direction-coded image, elements of A(x,y) are designated as code ϕ when $|\mathbf{V}| < \theta$ and where θ is a small threshold value. That is, code ϕ represents no brightness change. This is because the noise involved in the image has a large effect on the value A(x,y). Figure 1 is a schematically drawn image in which the picture elements where $A(x,y) \neq \phi$ are represented by unit vectors with arrows and the picture elements where $A(x,y) = \phi$ by blanks.

In the strict sense of the independence of the direction-coded image A(x,y) from the contrast or level change in a brightness image g(x,y), the threshold value θ should be a floating type allowing the value θ to be controlled by a certain macroscopic value derived from the image data. In fact, in actual images in which the delicate grey level differences dominate, the subsequent processing sometimes becomes inadequate. To compensate for this situation, it is desirable to determine the threshold value θ as follows:

$$\theta = \delta \iint_D |\mathbf{V}(x,y)| \, dx \, dy \quad , \qquad (3)$$

where δ is a small positive value and D is either a part of or the entire image area depending upon the application. Thus, θ becomes a floating value which is proportional to the average contrast for the area D. Consequently, the entire g(x,y) image set is transformed into the identical image A(x,y).

2.2 Transforming into a direction-coded image

In order to transform the brightness image g(x,y) into a direction-coded image by digital computer, the following calculation is adopted for the digital image. Here, the differential values in eq. (2) are replaced by the differences using adjacent 3x3 picture elements as shown in Fig. 2(a):

$$\left.\frac{\partial g}{\partial x}\right|_{i,j} = \sum_{k=-1}^{1} g(i+1,j+k) - \sum_{k=-1}^{1} g(i-1,j+k) \qquad (4)$$

$$\left.\frac{\partial g}{\partial y}\right|_{i,j} = \sum_{k=-1}^{1} g(i+k,j+1) - \sum_{k=-1}^{1} g(i+k,j-1) \quad .$$

In our computer simulation, a digital image of 256x256 picture elements, each of which was of 6-bit brightness (64 levels), was used to yield a direction-coded image whose picture elements consisted of 6-bit direction values(64 directions) as shown in Fig. 2(b). In this image, the numbers 0 through 63 are assigned for the 64 directions, and the number 64 is assigned for ϕ

If the threshold value θ is too small, the direction-coded image A(x,y) is not unique as it is affected by the noise involved in the image. If θ is too large, even the necessary information will be eliminated. Therefore, an adequate threshold value should be carefully chosen or automatically controlled by the image itself. In Fig.3 the total number of picture elements whose values are not ϕ is plotted for an actual laboratory scene by changing the threshold value θ as a parameter. A steep increase is seen in the curve where θ is less than 20 . This is mainly due to the noise existing in the brightness image. From this figure, it can be said that the optimum threshold value is, in general, somewhere between 15 and 20, although it of course depends on the imaged objects and the characteristics of the imaging device used.

Special-purpose hardware for transforming a brightness image into a direction-coded image can be constructed as shown in Fig. 4. In this construction, the threshold value is determined by the image value in the previous scanning frame of the TV camera by using a modified version of eq. (3). This threshold value holds for the present frame period. Thus, the direction-coded image can be obtained in a real time mode by synchronizing it with original input brightness image from a TV camera.

3. Recognition Algorithm of a Parallelepiped

3.1 Principles of recognition

Among actual three-dimensional objects, there are quite a number of box-like objects, since factories make extensive use of boxes in packing finished products for shipment. Therefore, from the viewpoint of the actual application of scene analysis techniques, especially to physical distribution systems, it is extremely important to refine the recognition algorithm of parallelepipeds.

For simplicity, the recognition algorithm of a box-like object viewed from the top is first described. In the direction-coded image, each picture element represents the tangential direction of brightness change. Therefore, quite a few picture elements with almost the same direction code are arranged on the linear edge line of an object, as long as a brightness difference exists on the edge line. Figure 5(a) shows a simple example of a rectangular shape, which corresponds to the top view of a box-like object. In this case, the frequency distribution of each direction code gives four peaks for every angle difference, as shown in Fig. 5(b). Therefore, when the four values at $\alpha + \frac{\pi}{2} i$ $(i=0,1,2,3)$ are added, the distribution curve gives a higher peak a $\alpha = \alpha_0$ shown in Fig. 5(c). This angle α_0 corresponds to the principal direction of the rectangle, namely the posture of the box-like object.

Even when some other figures like triangles or circles also exist in the same image as a background, the position of the highest peak in the frequency distribution curve in Fig. 5(c) is not usually affected, as the directions of the edge lines of these figures do not possess a pitch of $\frac{\pi}{2}$. The situation is quite the same when there are some characters or markings on the rectangular surface. Therefore, it is possible to accurately detect the orientation of the rectangle from the peak point in the frequency distribution curve of the direction-coded image.

The next step is to eliminate all direction codes except the neighbors of $\alpha_0 + \frac{\pi}{2} i$ $(i=0,1,2,3)$. Thus, the edge lines of the rectangle are enhanced because all other background noise and figures are considerably reduced. This processing may be called "direction code filtering". Further processing is done as shown in Fig. 5(d). That is, the picture elements with direction codes of α_0 and $\alpha_0 + \pi$ are projected on the line L whose angle is $\alpha_0 + \frac{\pi}{2}$, and are counted to yield a frequency curve q. Then, the peaks in the frequency distribution are detected, and the stripe regions are generated so that each region contains an edge line. The regions for the remaining two edge lines whose direction codes are $\alpha_0 + \frac{\pi}{2}$ and $\alpha_0 + \frac{3}{2}\pi$ are also detected in the same manner. These regions are then used to filter out all other picture elements, thus enabling the edge lines to become more pronounced. After this filtering operation, the determination of the edge lines becomes much easier, and the ordinary method of line determination can be easily applied. Examples of these processes are shown in Fig. 6.

3.2 Recognition of obliquely imaged parallelepipeds

When a parallelepiped placed on a table is viewed from above at an oblique angle, the periodicity of the $\frac{\pi}{2}$ angle differences among the edge lines in the image disappears. However, by extending the above-mentioned algorithm under the following three premises, the recognition of a parallelepiped becomes possible:

(1) The imaging device is placed far from the object compared to the size of the object so that the distortion of the image caused by projection becomes negligible.
(2) The object is placed on a horizontal plane.
(3) The angle J between the vertical line and the optical axis of the imaging device must be known.

Under these circumstances, edge lines with angles of $\alpha + \frac{\pi}{2} i$ $(i=0,1,2,3)$ in the image at $\gamma = 0$ are transformed to edge lines with angles of β_i $(i=0,1,2,3)$ when viewed obliquely at $\gamma = \gamma$, as shown in Fig. 7(a) and 7(b). The relationships between α and β_i are as follows:

$$\tan \beta_i = \tan (\alpha + \frac{\pi}{2} i) \cos \gamma \qquad (5)$$

where i = 0, 1, 2 and 3.

Therefore, by calculating β_i from eq.(5) with α as a parameter, the number of picture elements $N(\beta_i)$ at angle β_i are counted from the direction-coded oblique image. Then, the frequency distribution curve $N(\beta_0)+N(\beta_1)+N(\beta_2)+N(\beta_3)$ is obtained as shown in Fig.7(c).

The angle $\alpha = \alpha_0$ at which the frequency distribution curve gives a maximum peak corresponds to the orientation of the object when viewed vertically from the top ($\gamma = 0$). Therefore, by using this angle α_0, β_i is computed from eq.(5) for $\alpha_0 + \frac{\pi}{2} i$. Next, the filtering operation on the direction-coded image extracts only the picture elements whose directions coincide with β_i, $\frac{\pi}{2}$ and $\frac{3}{2}\pi$. This enhances true edge lines of the obliquely imaged parallelepiped considerably. Examples of these processes as well as subsequent edge detection and determination processes are shown in Fig. 8. By this method, it becomes possible to accurately detect edge lines without the confusion normally introduced by a noisy background or complex pattern information on the object surfaces.

4. Recognition Algorithm of Surface Information

4.1 Spatial transformation

To read and recognize surface information on an obliquely imaged parallelepiped, the image should first be converted to the normalized image having no relation to the position and orientation of the object. This can be effected by adequate expansion, contraction, shift and rotation.

As shown in Fig. 9 , the lozenge-shaped image portion corresponding to the upper surface of the object is transformed into a square image so that the edge points A, B, C and D correspond to the edge points A^1, B', C and D' respectively This is done by linear coordinate transformation by neglecting the projection distortion of the imaging system. That is, there exists the following relationship between the pre-transformed coordinates (x,y) and the post-transformed coordinates (X,Y) ;

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{w} \begin{pmatrix} C_x - A_x & B_x - A_x \\ C_y - A_y & B_y - A_y \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix} + \begin{pmatrix} A_x \\ A_y \end{pmatrix} , \quad (6)$$

where A_x , Ay ,..., Dx $ Dy denote the x and y coordinates of the edge points A, B, C, and D in the oblique image, and w corresponds to the length of the edge line of the square in the transformed image. From this equation, a picture point (X,Y) in the new image is obtained from the original image by averaging the picture element data at (x,y) or its vicinity, or by other interpolating methods. Figure 10 shows an example of this spatial transformation, in which the image is transformed as if it were viewed from the top of the object. In this normalized square image, the number of picture elements is standardized at 64x64. As this image is normalized according to its position and the size of its surface, a pattern matching method with standard patterns is applicable. This will be further covered in the next section.

In the above example, a method of transforming an original brightness image into a normalized brightness image of the surface is described. However, it is also easy to convert directly from the direction~coded oblique image into the direction-coded normalized square image of the surface. That is, the direction code at (X,Y) is computed as follows: First, the coordinate (x,y) is computed from eq.(6) , and the direction code β at (x,y) is determined from the direction-coded original image by a suitable averaging or interpolating method. Then, the direction code β is converted into α using eq*(5) . Finally, the principal directiα_0 (see Fig. 5(a) } is subtracted.

4.2 Information compression and recognition

The two-dimensional normalized image of the surface discussed in the above section is still too redundant to directly perform pattern matching with standard patterns. Also, in this case, the large volume of image data would necessitate an enormous memory capacity for standard patterns. Therefore, the following data compression is performed:

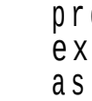
(1) From the direction-coded, normalized square image of the surface, picture elements with a certain direction code are extracted and projected on the line L perpendicular to the direction code to yield the frequency distribution curve q as shown in Fig. 11(a) .
(2) This frequency distribution curve is transformed into the Fourier series and the magnitudes of the lower four harmonics (from 1st through 4th) are computed.
(3) The above two processes are performed for all of the direction codes, thus giving a data table with 64 directions times 4 harmonics as shown in Fig. 11(b) .
(4) The data table thus obtained shows how the four magnitudes of the lower harmonics change with the direction change. Therefore, by regarding the four rows of the table as four waveforms, these can be transformed into the Fourier series.
(5) The lower eight harmonics (from 1st through 8th) are chosen to compute each magnitude, thus giving a final table with 8x4 magnitude data as shown in Fig. 11(c) .

By the above data compression, a normalized image is expressed as information with 8x4 (=32) words. In other words, an image of the surface corresponds to a point in a 32-dimensional space. Therefore, the distances in the 32-dimensional space between the unknown image pattern and the pre-determined standard patterns are computed. The pattern with the minimum distance is determined as the best match.

The algorithm mentioned above has the following features; it is (1) invariant to the pattern rotation, and (2) invariant to the pattern shift. These features arise from the following facts:

(1) The frequency distribution of each direction code is transformed into the Fourier series.
(2) The magnitude change of each harmonic in relation to the direction change is transformed into the Fourier series.
(3) Only the magnitude data are used, and the phase data as well as d.c. component are neglected.

Therefore, even when the spatial transformation in Fig. 10 gives an upside-down image, the subsequent matching

process gives the correct answer. For example, the symbols  and  as well as A with any arbitrary angle will lead to the same result . This characteristic is extremely useful when the posture of a pattern can not be specified beforehand. Table 1 shows an example of some experimental data which represent the relative distances in 32-dimensional space between some simple example patterns. It can be seen from the table that the same patterns give smaller distances regardless of the pattern inclinations.

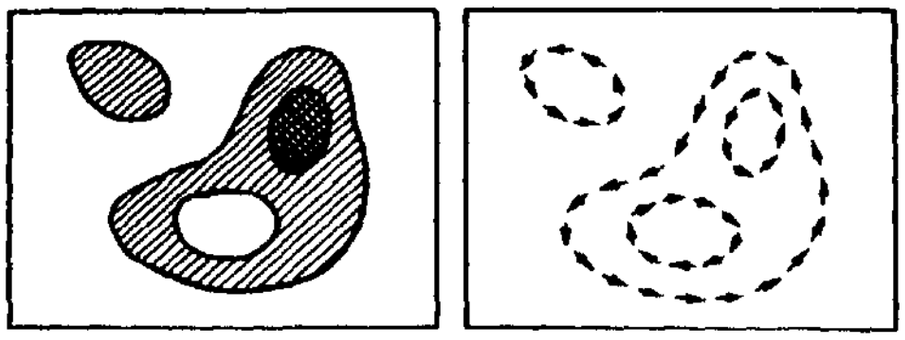
5. Conclusion

The algorithm discussed in this paper is summarized as follows. First, the edge lines of a parallelepiped are extracted based on the direction-coded image. Next, the position of the object is determined irrespective of the existence of a noisy background or the existence of surface information of considerable complexity. Then, by using the edge point data of the detected object, the surface image is projected as a normalized square image as if it were being viewed from a direction perpendicular to the surface. Finally, the double Fourier transformation is performed on the direction-coded image of the normalized square image. One of the Fourier transformations is in conjunction with the axial direction in order to find the distribution of the direction code, and the other is with radial direction to determine magnitude variation. As a result, an image is largely reduced as to its quantity of information and is transformed into position-invariant and rotation-invariant data for easier pattern matching.

The main characteristics of this method arise from the fact that the direction-coded image is preferentially used. It is possible to regard this direction-coded image as the image in which the microscopic features in the original brightness image are distributed. For these microscopic features of the original image, macroscopic treatment such as frequency distribution calculation and Fourier transformation is performed. Therefore, this method is hardly effected by background noise or local noise which sometimes exists in the image. Such macroscopic processing of microscopic features is one of the most effective methods of actually applying image processing to industrial use. Furthermore, this method seems to hold promise of effective application in more complicated scene analysis and pattern recognition problems.

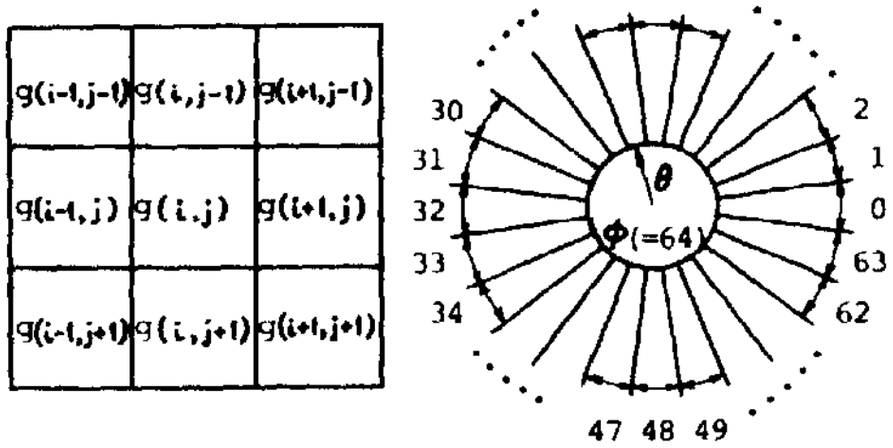
References

1. N.J.Nilsson : "A Mobile Automaton : An Application of Artificial Intelligence Techniques", Proceedings of the 1st IJCAI, Washington D.C., 1969
2. J. Feldman, K. Pingle et al. : •'The Use of Vision and Manipulation to Solve the Instant Insanity Puzzle", Proceedings of the 2nd IJCAI, London, 1971
3. H.G.Barrow, R.J.Popplestone : "Relational Descriptions in Picture Processing", Machine Intelligence, Vol.6, pp.337-396, Edinburgh Univ. Press, 1971
4. M.Ejiri, T.Uno, H.Yoda et al. : " A Prototype Intelligent Robot that Assembles Objects from Plan Drawings", IEEE Trans, on Computers, Vol.C-21, No.2, February 1972.
5. H.Yoda, S.Ikeda, M.Ejiri : "A New Attempt of Selecting Objects Using a Hand-eye System", Hitachi Review, Vol.22, No.9,1973
6. M. Ejiri, T.Uno et al,: "A Process for Detecting Defects in Complicated Patterns", Computer Graphics and Image Processing, Vol.2,No.3/4, 1973
7. T.Uno, S.Ikeda, J.Motoike, M.Ejiri et al.: "An Industrial Eye for Recognizing Shapes and Positions of Specified Objects" (in Japanese), No. PRL74-29, The Institute of Electronics and Communication Engineers of Japan, 1974
8. F.O'Gorman, M.B.Clowes : "Finding Picture Edges through Collinearity of Feature Points", Proceedings of the 3rd IJCAI, Stanford, 1973
9. J.K.Bowker : "Edge Vector Image Analysis", Proceedings of the 2nd Int. Joint Conf. on Pattern Recognition, Copenhagen, 1974



(a) g(x,y) (b) A(x,y)

Fig. 1 Schematic drawing of a brightness image (a) and a direction-coded image (b).



(a) adjacent 3x3 picture elements for calculating gradient vector

(b) area partitioning of a vector plane for direction coding

Fig. 2 Direction coding

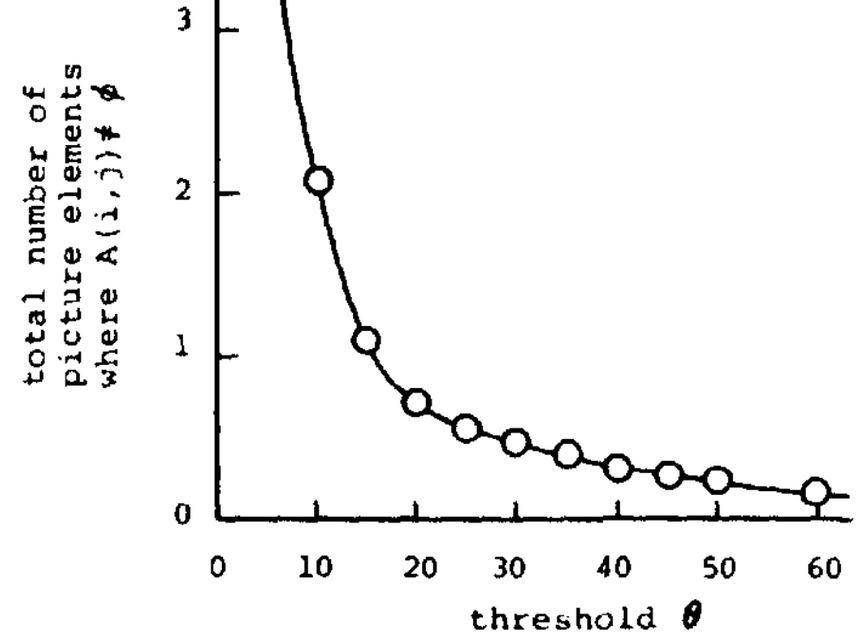


Fig. 3 Change of total number of coded elements to threshold change.

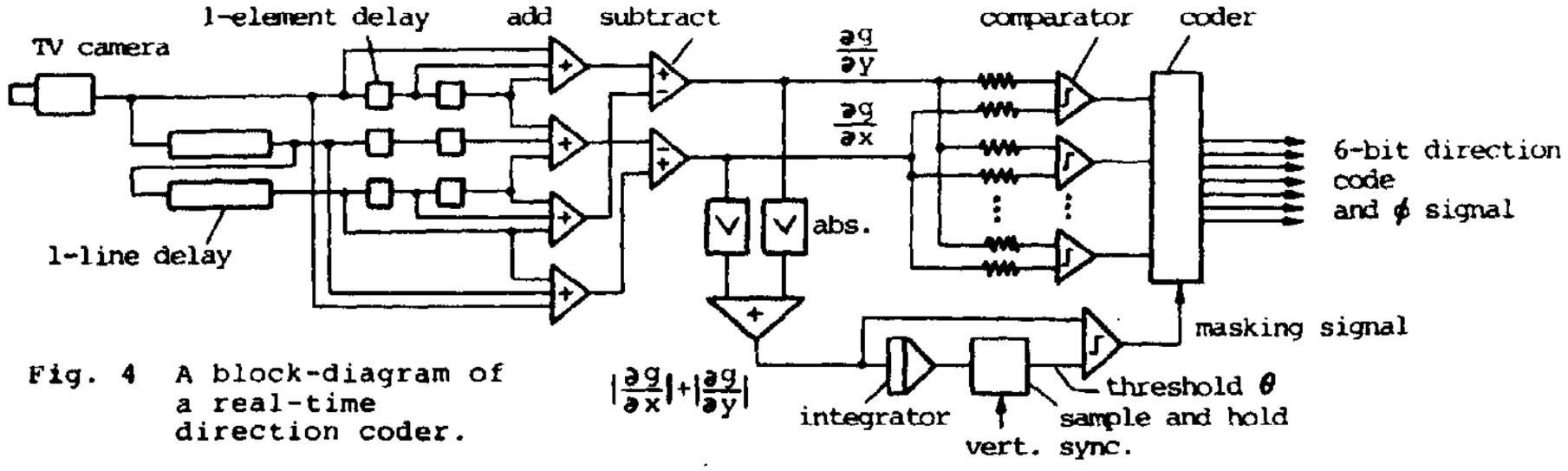
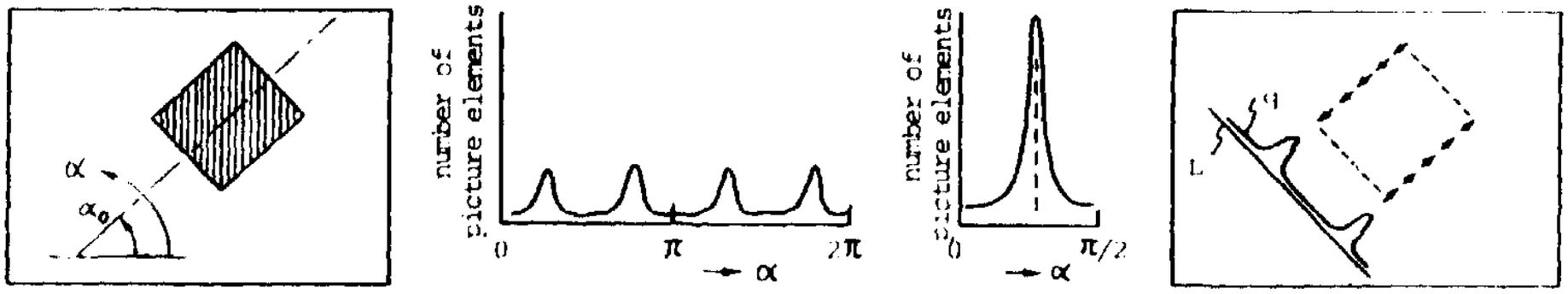


Fig. 4 A block-diagram of a real-time direction coder.



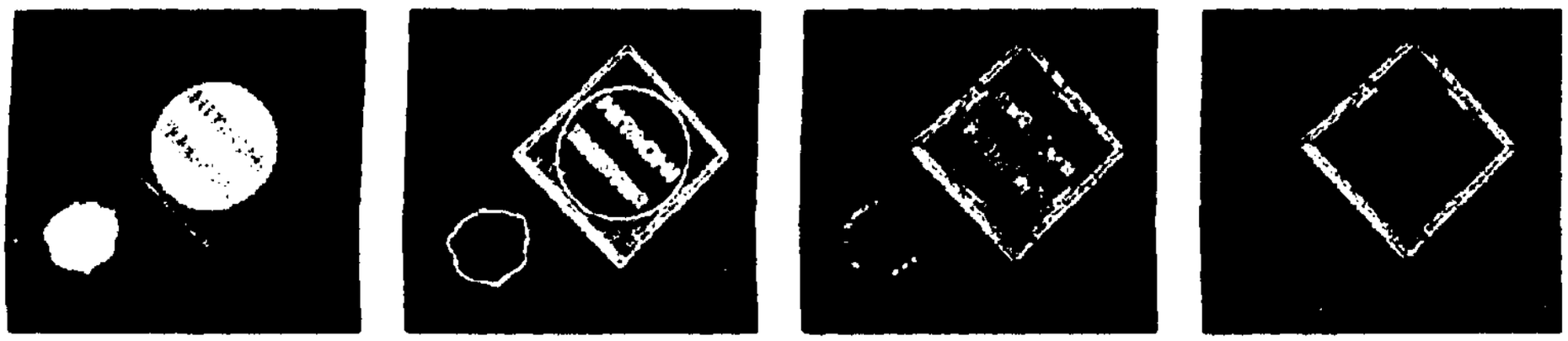
(a) an image of a rectangle

(b) frequency distribution

(c) frequency distribution with max. $\pi/2$

(d) edge line extraction by distribution finding

Fig. 5 Orientation determination of a rectangular shape.



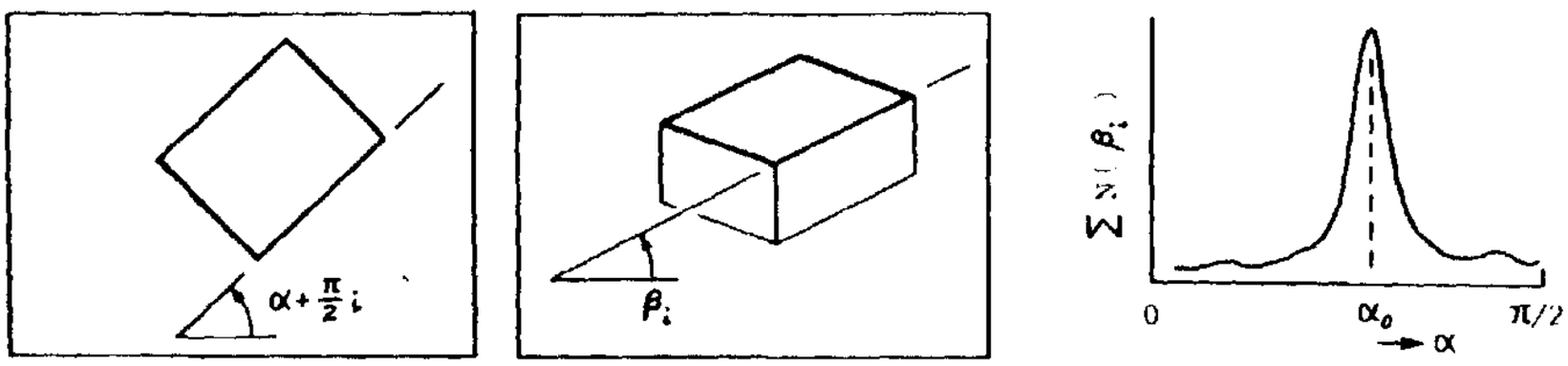
(a) original brightness image

(b) direction-coded image, all picture elements where $A(i,j) \neq \phi$ are plotted uniformly.

(c) direction-coded image after "direction code filtering"

(d) direction-coded image after "regional filtering"

Fig. 6 Examples of rectangle recognition.



(a) top view of a parallelepiped ($\gamma = 0$)

(b) oblique view of a parallelepiped ($\gamma \neq \gamma$)

(c) frequency distribution

Fig. 7 Orientation determination of a parallelepiped.

(a) original brightness image

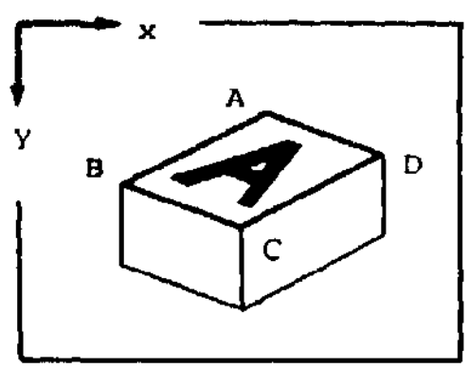
(b) direction-coded image, all picture elements where $A(i,j) \neq \phi$ are plotted uniformly.

(c) after "direction code filtering"

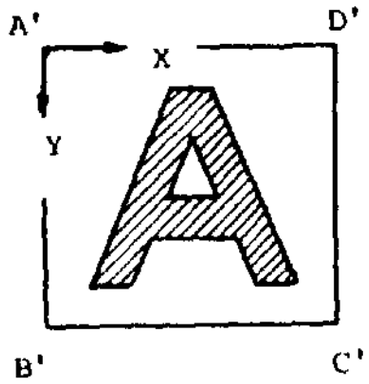
(d) edge line detection

(e) edge line determination

Fig. 8 Examples of parallelepiped recognition.

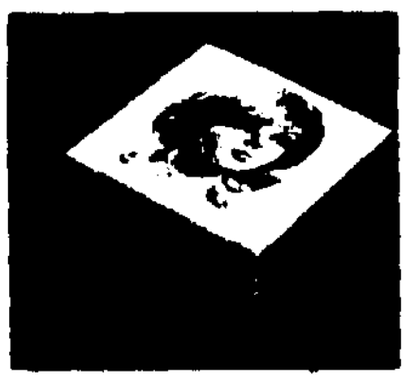


(a) original image



(b) normalized, square image

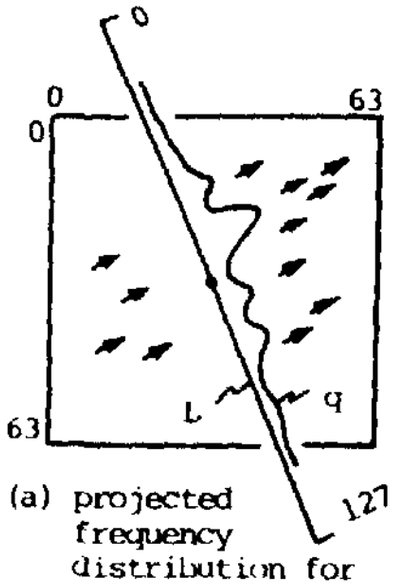
Fig. 9 Spatial transformation.



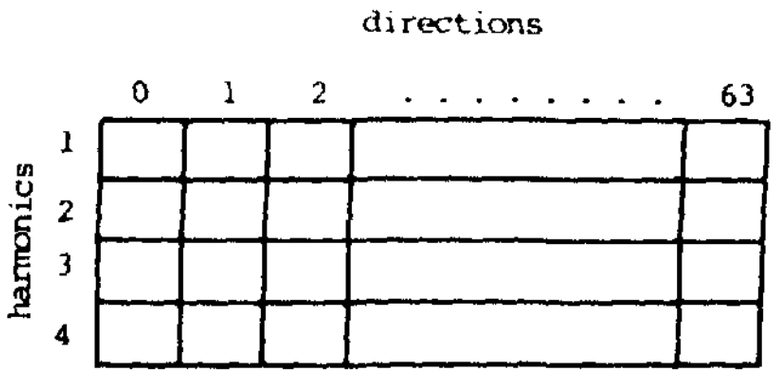
(a) original image

(b) normalized, square image

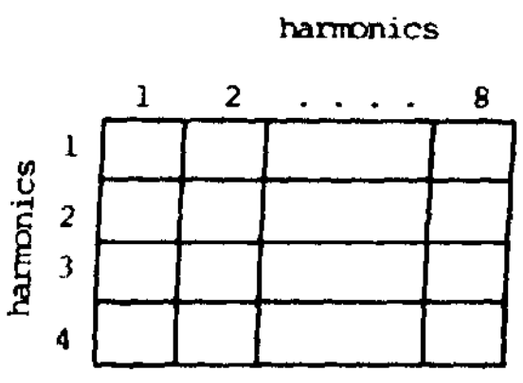
Fig.10 An example of spatial transformation.



(a) projected frequency distribution for each direction code.



(b) data table of magnitudes of harmonics.



(c) final data table for pattern matching.

Fig.11 Data compression

Table 1

Experimental results of relative distances between sample patterns.

	20	866	1385	666	643	664	700
	900	56	2080	822	871	867	852
	1379	2095	236	1768	1748	1753	1746
	696	810	1812	214	263	225	224