Real-Time Imaging **5**, 63–74 (1999)
Article No. rtim.1998.0135, available online at http://www.idealibrary.com on IDEAL

Fast Reflectional Symmetry Detection Using Orientation Histograms

A simple and fast reflectional symmetry detection algorithm has been developed in this paper. The algorithm employs only the original gray scale image and the gradient information of the image, and it is able to detect multiple reflectional symmetry axes of an object in the image. The directions of the symmetry axes are obtained from the gradient orientation histogram of the input gray scale image by using the Fourier method. Both synthetic and real images have been tested using the proposed algorithm.

C. Sun[1] and D. Si[2]

[1] *CSIRO Mathematical and Information Sciences, Locked Bag 17, North Ryde, NSW 2113, Australia*
E-mail: changming.sun@cmis.csiro.au
[2] *Department of Electronics Engineering, Beijing University of Aeronautics and Astronautics Beijing 100083, China*

Introduction

It has been proposed that symmetry plays a major role in object recognition by providing a canonical axis for the representation of shapes [1]. Many objects around us are strongly constrained. For instance, many cultural artifacts and natural objects are reflectionally (bilaterally or mirror) symmetric. A reflectional symmetry has a reflection line, for which the left half space is a mirror image of the right half. One of the goals of an image understanding system is to identify and locate a specified object in the scene. In such cases, the system must have some knowledge of the shape of the desired object. Symmetries are good candidates for describing shapes. It is a powerful concept that facilitates object detection and recognition in many situations. These representations can be used in robotics for recognition, inspection, grasping, and reasoning.

Most of the work carried out on symmetry detection has been based on edge, contour or point set information. Burton *et al*. [2] considered a simple indexing scheme to implement the exponential pyramid data structure for particular symmetries. Wolter *et al*. [3] described exact algorithms for detecting all rotational and involutional symmetries in point sets, polygons and polyhedra. Atallah [4] and Davis [5] used evaluation techniques for symmetry detection on images composed of line segments, circles and points. Highnam [6] presented an asymptotically optimal algorithm to locate all the axes of mirror symmetry and optimal algorithms for finding rotational symmetries of a planar point set. Marola [7] presented an algorithm for finding the number and position of the symmetry axes of a symmetric or almost symmetric planar image. This method required the evaluation of some rational functions. He also presented a recognition procedure based on the measurements of the degree of symmetry of planar intensity images by superposition or by convolution [8]. Zabrodsky *et al*. [9] defined a Continuous Symmetry Measure to quantify the symmetry of objects. They also presented a multi-resolution scheme [10] that hierarchically detects symmetric and almost symmetric pat-

1077-2014/99/010063 + 12 $30.00

terns. Parry-Barwick and Bowyer [11] developed methods that can detect both hierarchical and partial symmetry of two-dimensional (2D) set-theoretic models with components constructed with a few straight edges or polynomials. This method had the disadvantage of being computationally intensive. Zielke *et al.* [12, 13] only looked at vertical or near vertical symmetry axes in an image for car-following applications. Masuda *et al.* [14] described a method of extracting rotational and reflectional symmetry by performing a correlation with the rotated and reflected images. But the method had a high computational cost and memory requirements. Bowns and Morgan [15] presented a method of extracting facial features using natural orientation information. But this method is not suitable for general shaped objects when the peak orientation is not related to the orientation of the symmetry axis. O'Mara and Owens [16] used the directions of the principal axes of an object as the initial values of the symmetry axis. Sun [17] used a direct correlation method on the orientation histogram to obtain the most likely symmetry axis.

In this article we investigate the use of gradient information for symmetry detection in a gray scale image by using the Fourier technique. Our purpose for using the Fourier method is to increase the computation speed further compared with the method used in [17]. We also try to find multiple symmetry axes in an object rather than just obtain one of the most likely symmetry axes, as in [17]. The following sections describe the gradient orientation histogram, give the algorithm for finding the orientations and position of the symmetry axes, and show the results of the algorithm on both synthetic and real images. Comparison of several existing methods and the method developed in this paper is also described. The final section provides some conclusions.

Gradient Orientation Histogram

The image formation includes imaging geometry, surface photometry, and surface contours. The image itself is often ambiguous in its representation of scene information, and the interpretation of such an image requires a combination of analytical tools and knowledge or assumptions about the scene (objects) and the image production [18]. The observed brightness of objects corresponds to the image irradiance. For our analysis, we will only consider Lambertian surfaces. Neither specular reflection nor shadowing effects will be treated here. For an object surface described by $I(x,y)$, the surface gradient vector $[I_x, I_y]^T$ is defined by:

$$I_x = \frac{\partial I(x,y)}{\partial x}, \qquad I_y = \frac{\partial I(x,y)}{\partial y}. \tag{1}$$

The orientation of this gradient vector is:

$$\phi = \arctan(I_y/I_x). \tag{2}$$

The domain of ϕ is $[0, 2\pi)$. It is expected that the gradient orientation information of an image can be used for symmetry detection. Details of the method will be described in later sections. This method can be misled by the presence of a periodic texture within the object (if the texture pattern does not have the same symmetry axis with the object), as these repetitive patterns may give high peaks in the orientation histogram compared with those from the object's global shape. However, as long as the predominant gradient orientations are symmetric, the algorithm will be able to find the symmetry axes.

Figure 1 gives an illustrative shape of the gradient orientation histogram for a reflectionally symmetric object. The shape should be mirror symmetric, or nearly mirror symmetric, because of the digitization errors. Figure 2(a) shows a symmetric object, and Figure 2(b) is a circular gradient orientation histogram with the orientation angle starting from the center-to-right direction (0°) and increasing clockwise, and the radial length showing the number of pixels at this orientation within this image. The line passing through the center of Figure 2(b) indicates the symmetry orientation. It is easier for us to perceive the 2π periodicity of the gradient orientation histogram in this circular representation.

Orientation and Position of Symmetry Axes

Our algorithm is based on the gradient orientation distribution of the input gray scale intensity image. It

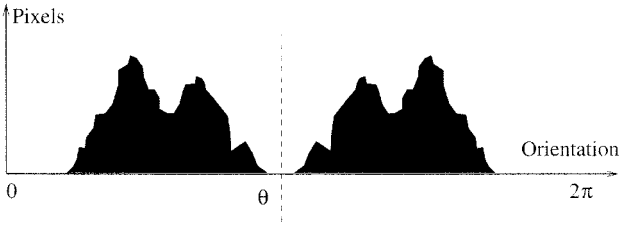


Figure 1. An illustrative shape of the gradient orientation histogram for a reflectionally symmetric object.

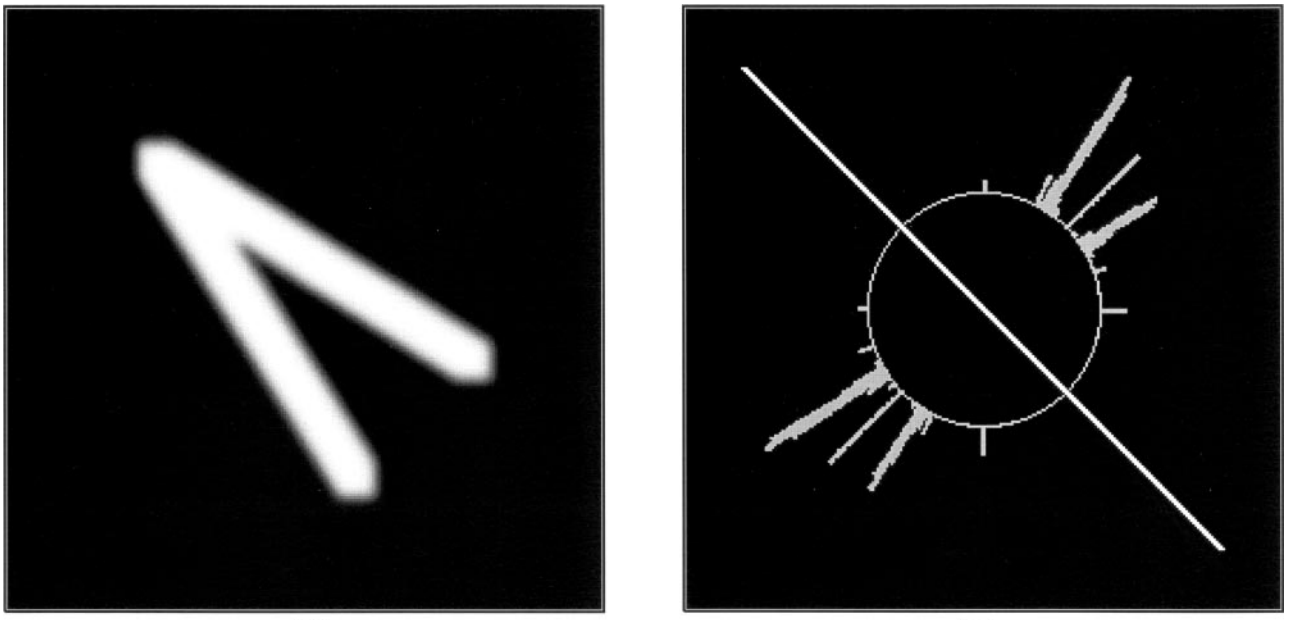


Figure 2. A symmetric object and its circular gradient orientation histogram. (a) An object with reflectional symmetry; (b) the circular gradient orientation histogram of the object with the symmetry orientation overlaid (the line passing through the center).

will also work for binary images in which the useful gradient information appears at the boundaries of objects. From the previous section, the histogram of this gradient orientation image can therefore be obtained [using Eqns (1) and (2)]. The range of the gradient orientation is from 0 to 360 degrees. Users are able to specify the number of bins for the histogram. In order to have better angular resolution, the bin number can be chosen to be some larger numbers than 360. It can be observed that for a symmetric object in the image, its orientation histogram is also symmetric. It is also clear that this histogram function is periodic with period 360 (or 2π). That is,

$$h(\theta) = h(\theta \pm 2n\pi) \qquad n = 0, 1, 2, \cdots \tag{3}$$

where $h(\theta)$ is the gradient orientation histogram of the input image, and θ is in $[0, 2\pi)$. Sun [17] used a window at position x of the histogram with length π from the left and π from the right to calculate the following function:

$$c(x) = \int_{\theta=0}^{\pi} h(x+\theta)h(x-\theta)d\theta \tag{4}$$

The orientation of the symmetry axis is the value x which gives the maximal value for $c(x)$. In his paper only one symmetry axis is chosen, by simply selecting the maximal value of Eqn (4). In practice, the integral in the above equation is approximated by a sum of the discretized orientation histogram $h(i)$, $i = 0, 1, \ldots, N-1$. The signal processing operation in Eqn (4) is a correlation between the segments of the gradient orientation histogram. Since portions of the histogram are mirror images, a reflection prior to a shift and multiply is carried out, as is done when performing a convolution. In this paper we use the Fourier method to obtain the symmetry information. The Fourier transform $H(u)$ of $h(i)$ as defined in Eqn (5) will be used to obtain the convolutions of two signals. In our case, the two signals are the same, i.e. $h(i)$.

$$H(u) = \sum_{i=0}^{N-1} h(i)\exp\{-j2\pi iu/N\}, u = 0, 1, \ldots, N-1 \tag{5}$$

For a reflectionally symmetric object, its gradient orientation histogram $h(i)$ will be reflectionally symmetric or nearly so at one or more orientation angles. The convolution of histogram $h(i)$ with itself will produce peaks at those symmetry angles. This convolution can be easily obtained by the Fourier transform. Based

on the following *convolution theorem*:

$$f(x) * g(x) \Leftrightarrow F(u) \times G(u) \tag{6}$$

the convolution in the space domain can be obtained by taking the inverse Fourier transform of the product $F(u)G(u)$. By doing so, the computational cost can be reduced to the order of $N \log_2 N$ rather than the order of N^2. In our case $F(u) = G(u) = H(u)$. Therefore, we have $h(i) * h(i) \Leftrightarrow H(u) \times H(u)$, and we can use the inverse Fourier transform of $H^2(u)$ for obtaining the convolution shown in Eqn (4), i.e.:

$$h(i) * h(i) \Leftrightarrow H^2(u) \tag{7}$$

We also extend the work of Sun [17] to the cases where there exist multiple symmetry axes about the object. If there are multiple symmetry axes, the convolution function should have multiple peaks. So by setting an appropriate threshold, multiple peak positions which correspond to the orientations of multiple symmetry axes can be obtained.

The accuracy of the orientations of the symmetry axes obtained by searching the peak positions in the convolution function can be improved by fitting a parabola function around the local region of each peak. The optimal positions of the peaks can be obtained analytically from the fitted parabola function.

After the directions of the symmetry axes have been obtained, it is also necessary to determine the position of these lines. The center of gravity of the image (or the object of interest) can be used for determining the position of the symmetry axes.

The operations for obtaining the symmetry axis directions and position are the following:

1. Perform a gradient operation on the image;
2. Obtain the gradient orientation histogram;
3. Search for the convolution peaks of the gradient orientation histogram by using the Fourier transform to obtain the orientations of symmetry axes;
4. Obtain sub-angular positional accuracy by fitting a parabola function, and finding the peak position analytically;
5. Perform symmetry check or evaluation about the obtained symmetry axes;
6. Obtain the position of this symmetry axis by using the center of mass of the object; and
7. Draw the symmetry axes based on the orientation and position information obtained.

The flow chart of the algorithm is shown in Figure 3.

Experimental Results

The results for the described algorithm on both synthetic and real images are given in this section. The gradient of an image is obtained by using a 5×5 Sobel filtering operation on the input gray scale image [19]. The filtering splits the kernels into 2×1 sub-kernels and iteratively calculates the responses. The results are two gradients for the x- and y-directions, respectively. Then the gradient orientation at this point of the image can be obtained by using Eqn (2). When both I_x and I_y are zero at a point in the image, the gradient at this point is treated as not defined. By having a set of bins in the range of (0, 360), the orientation histogram of the image can therefore be obtained. No initial smoothing was applied to the original image before the gradient operation, as the process of obtaining the orientation histogram has the effect of cancelling the noise contribution.

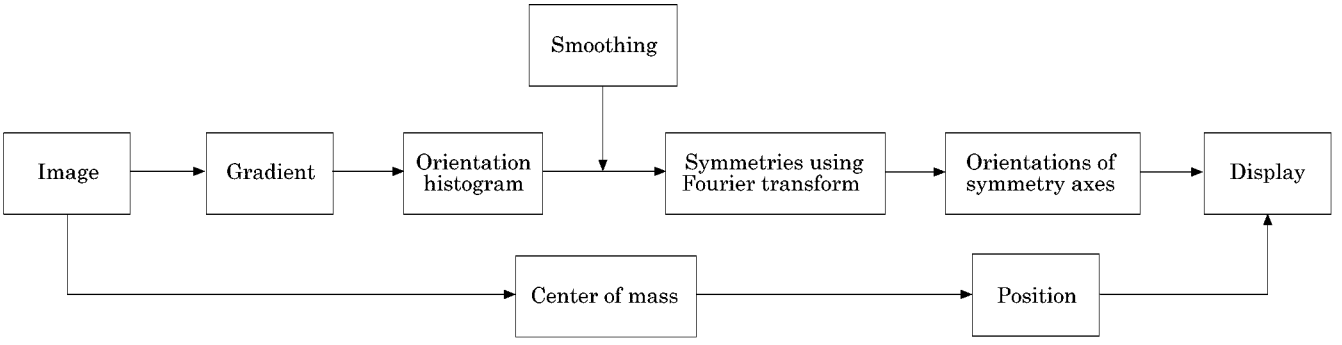


Figure 3. Flow chart of the algorithm for obtaining symmetry axes.

Because of the digitization effects, the boundary of an object consists of mostly zigzagged short line segments, and very often these line segments are either horizontal or vertical, with diagonal segments connecting the horizontal or vertical short lines. Therefore, in most of the gradient orientation histograms, peaks often appear at multiples of 45° (0°, 45°, 90°, 135°, 180°, 225°, 270° and 315°). This effect can be reduced by increasing the kernal size in the gradient operation to some extent. However, it is hard to eliminate the effect completely just by increasing the kernel size. The histogram is circularly smoothed with a median filter of size 5; that is, the smoothing window is wrapped around at the ends since the angular data are circularly continuous. The convolution peaks are found from the smoothed data. Only those peaks that are within 90% of the maximal value are chosen. The positions of the peaks are further refined by fitting a quadratic function around the local region (say five points) of each peak. The refined positions are obtained analytically by finding the maximal value of the quadratic function. Therefore, higher angular resolution can be obtained. The general form of the quadratic parabola equation is: $f(x) = a + b \cdot x + c \cdot x^2$. The maximum can be found where the slope is zero in the quadratic equation. The position of the peak can be found at $x = -b/2c$, as illustrated in Figure 4.

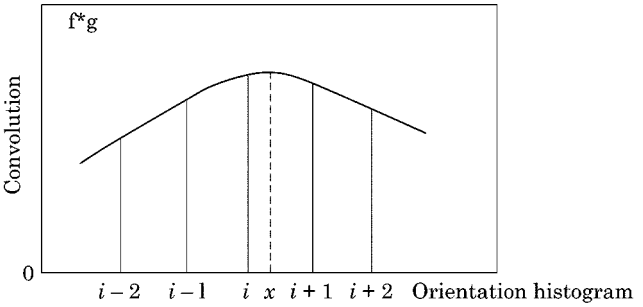


Figure 4. Sub-angular positional accuracy by finding the peak of a fitted quadratic parabola equation.

The histogram property is a necessary but not a sufficient condition for symmetry detection. That is, for certain non-symmetric objects, its orientation histogram might still be reflectionally symmetric. One example is shown in Figure 5. Figure 5(a) shows a non-symmetric object, while Figure 5(b) displays its idealized gradient orientation histogram. Although the orientation histogram is reflectionally symmetric, the object does not have the reflectional symmetry property. However, as long as we know that the object in the image is reflectionally symmetric, we can apply the algorithm to detect reflectional symmetries. If the condition is not sufficient, i.e. when the orientation histogram shows a reflectional symmetry but the actual object is not reflectionally symmetric, a further step might be necessary to check whether the object is

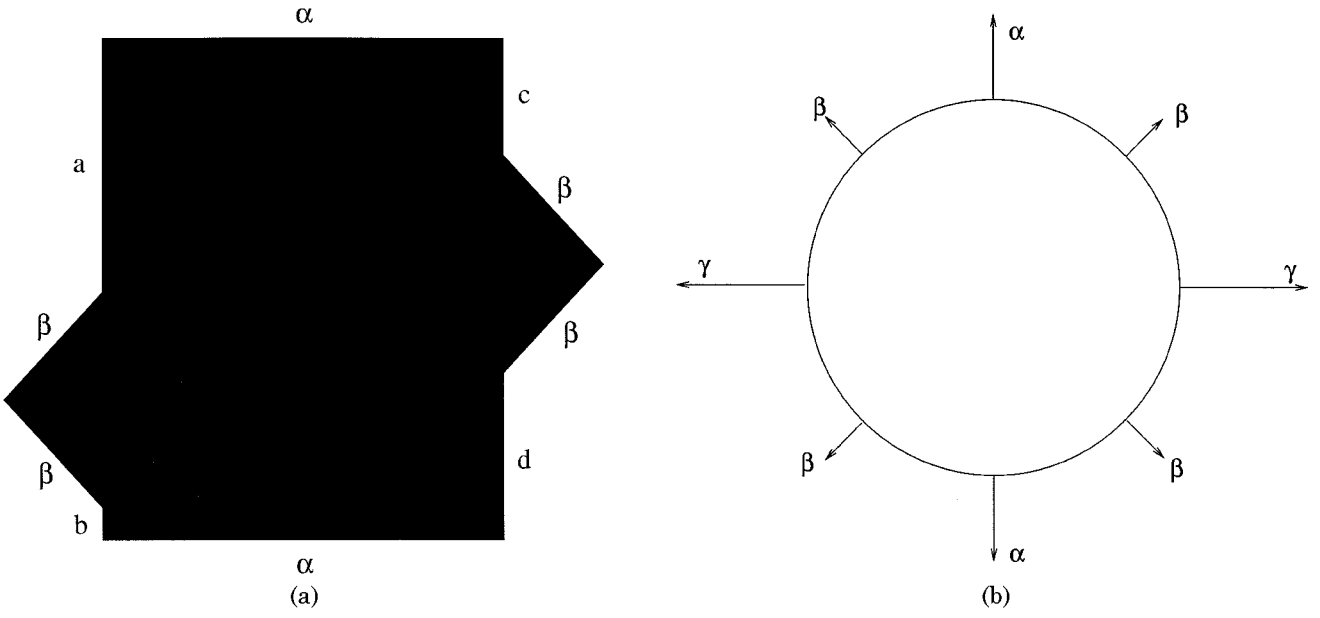


Figure 5. (a) A counter-example when the proposed algorithm might not work; (b) The ideal orientation histogram for (a) (where $\gamma = a + b = c + d$).

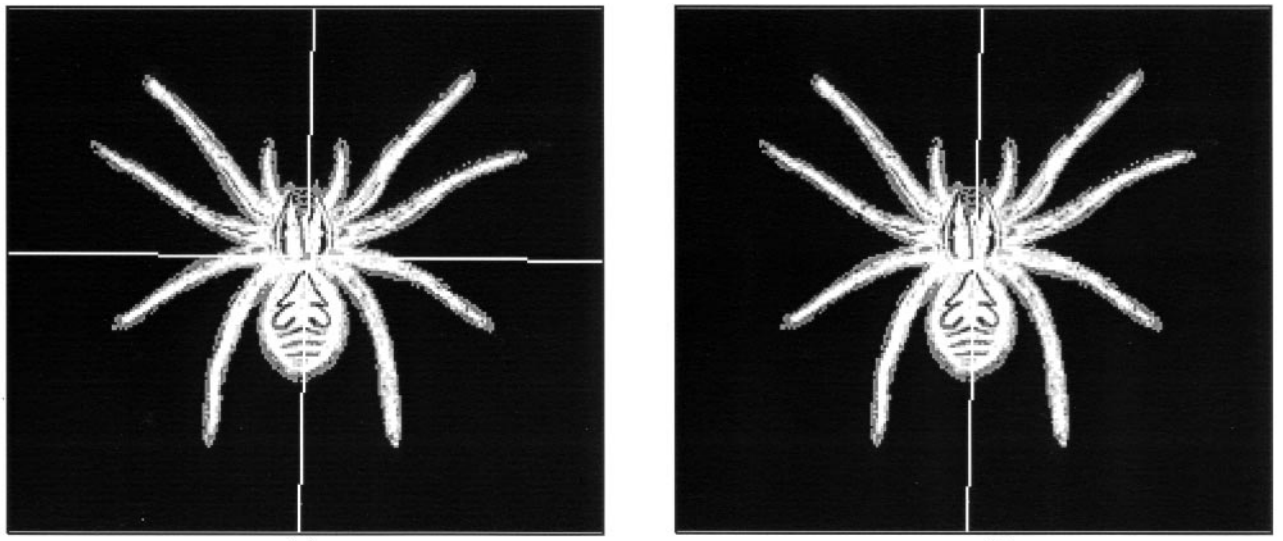


Figure 6. (a) Two symmetry axes were initially detected; (b) the incorrect symmetry axis was eliminated.

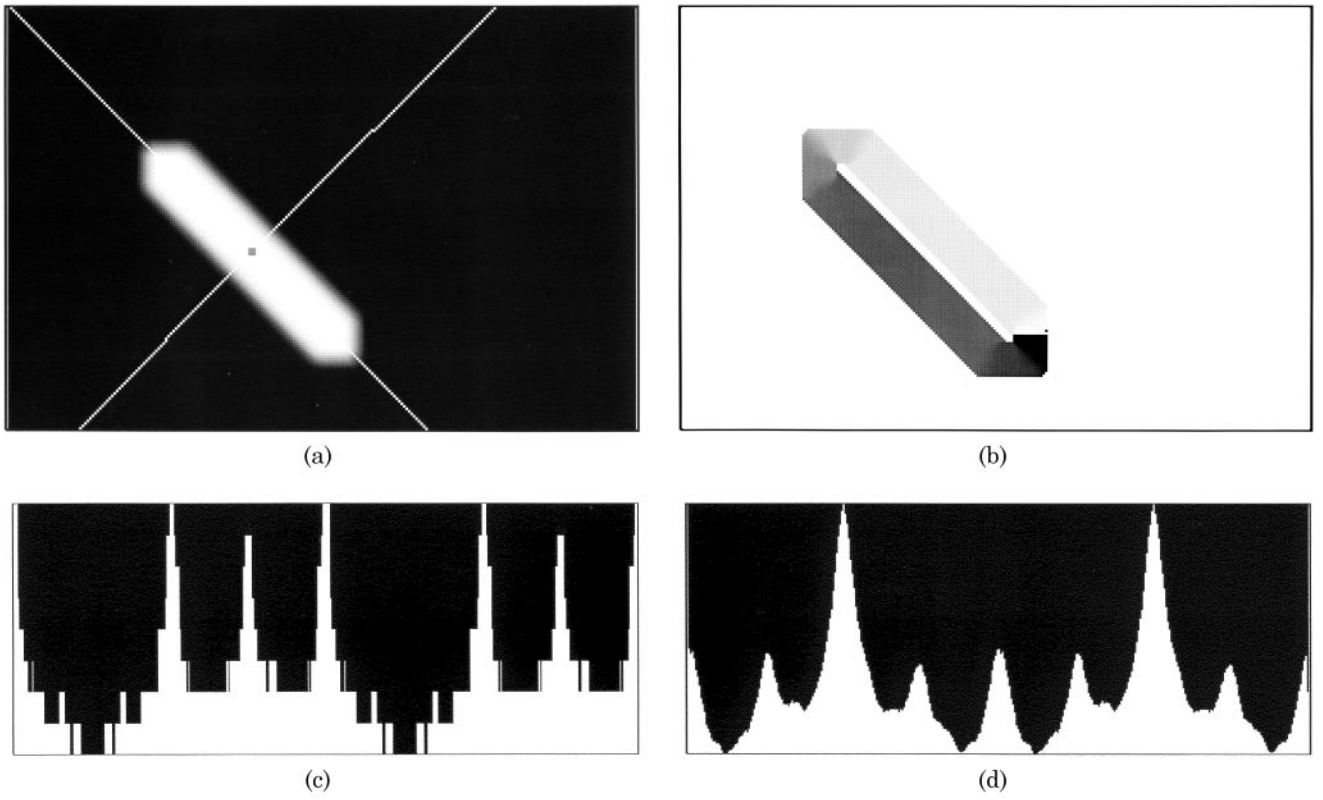


Figure 7. Symmetry detection result. Two symmetry axes have been detected by searching the high peaks in the convolution function. (a) Original image and the symmetry axes obtained; (b) gradient orientation image (normalized from 0−360 to 0−255); (c) median smoothed histogram of the gradient orientation image (b); and (d) the convolution function obtained using Fourier transform.

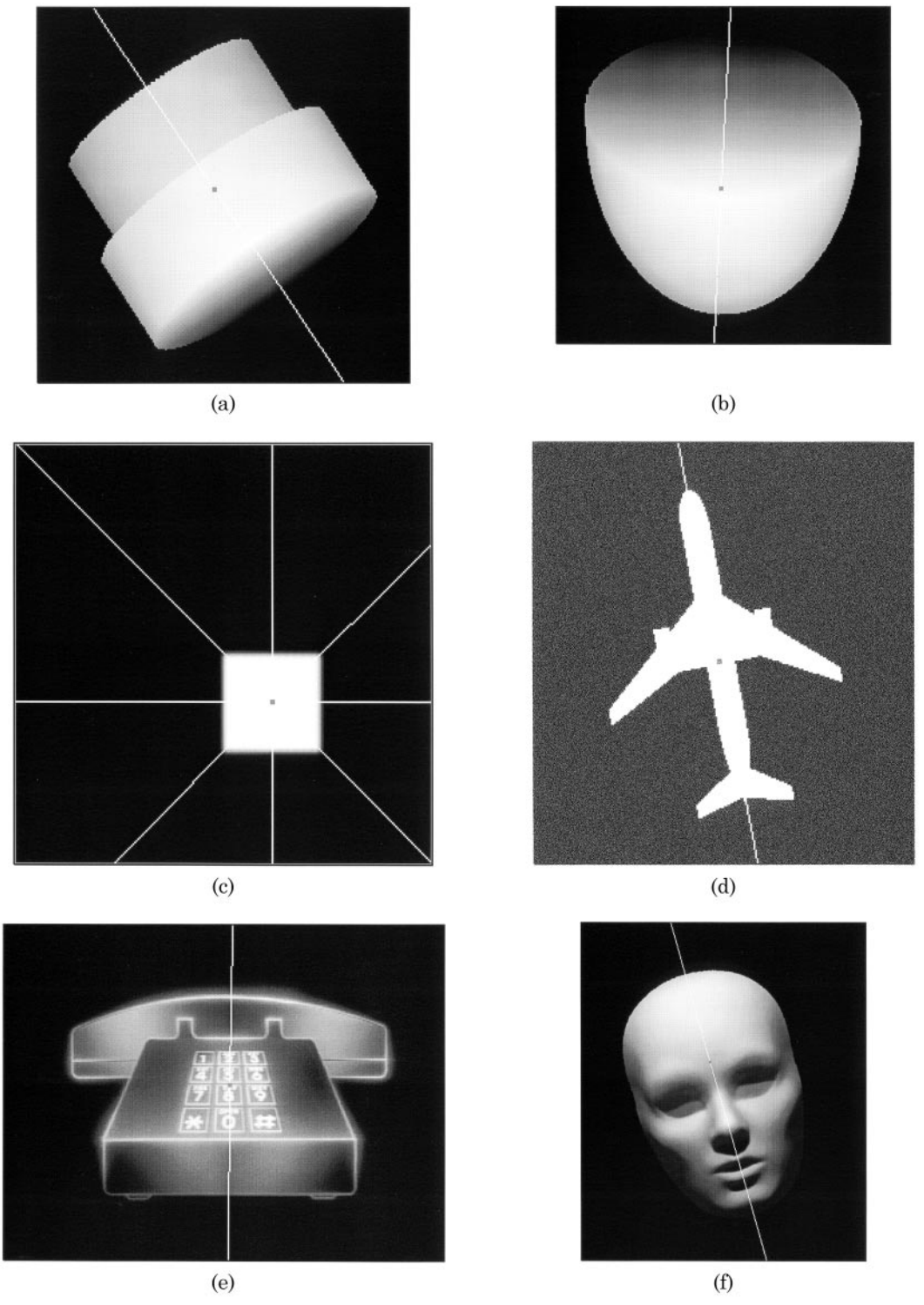


Figure 8. Symmetry detection results for several synthetic images. (a) A range image of an adapter; (b) a range image of a half sphere; (c) a square with four symmetry axes; (d) an aeroplane; (e) a telephone; and (f) a rotated human face.

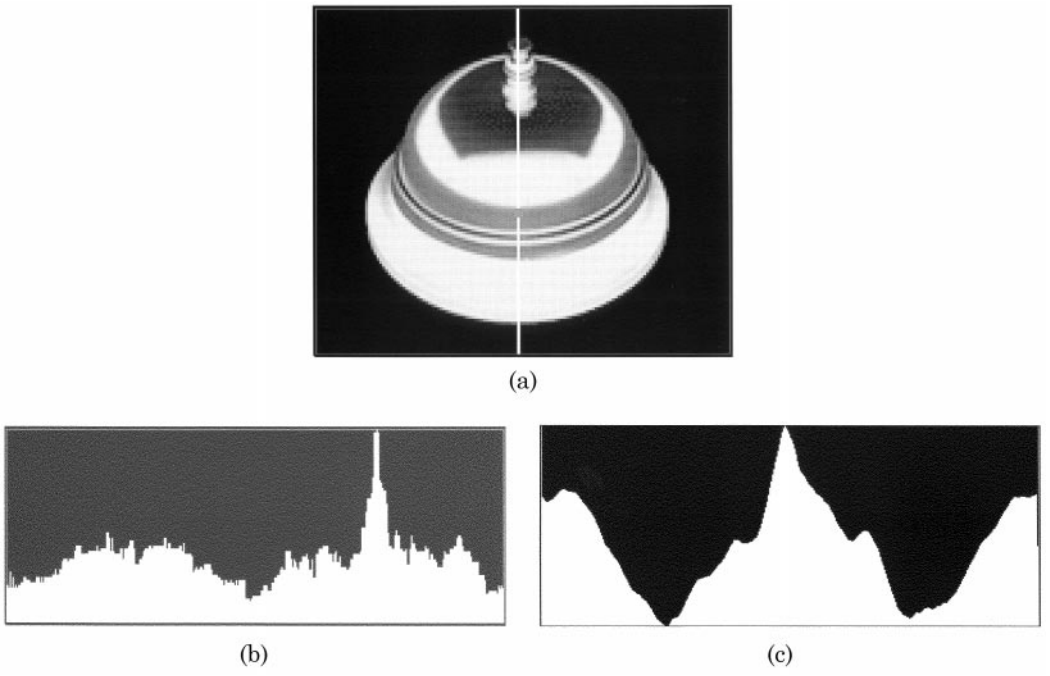


Figure 9. Symmetry detection. (a) an original image with the detected symmetry line overlaid; (b) the gradient orientation histogram; and (c) the convolution function obtained from Fourier transform.

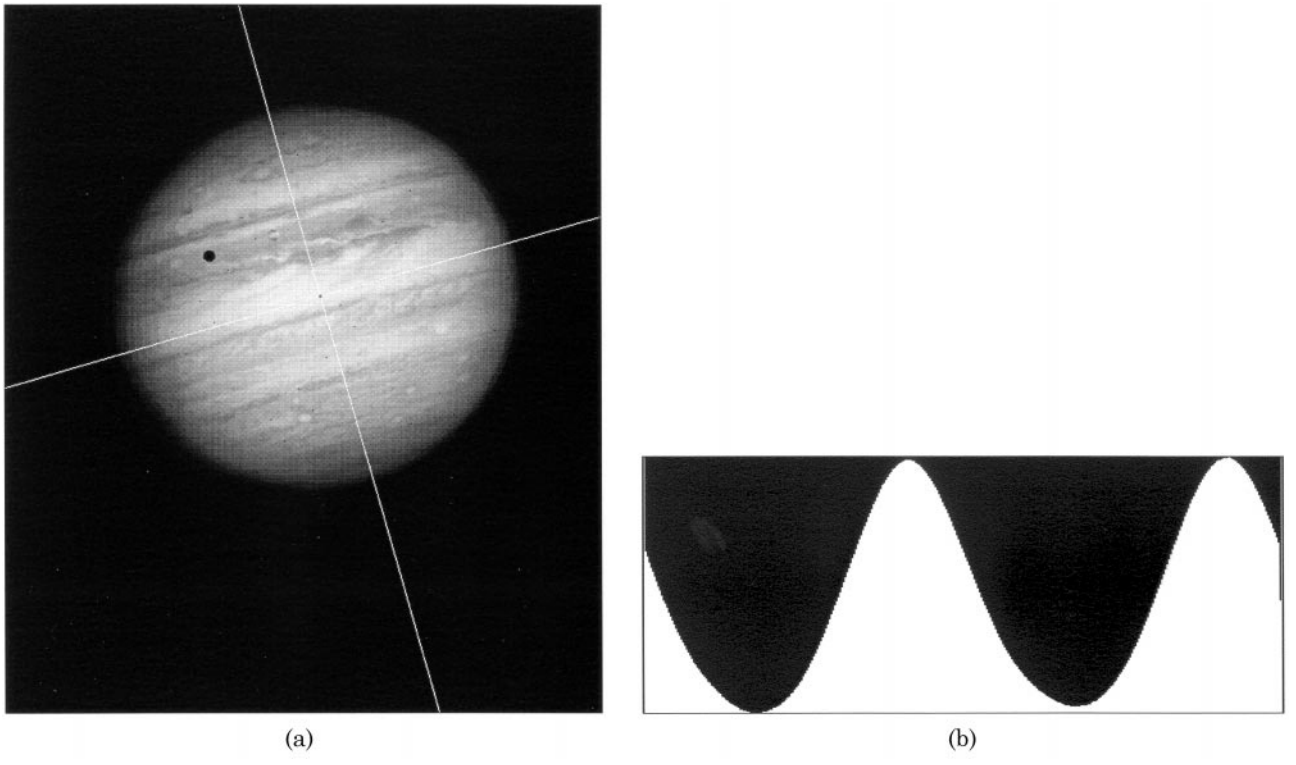
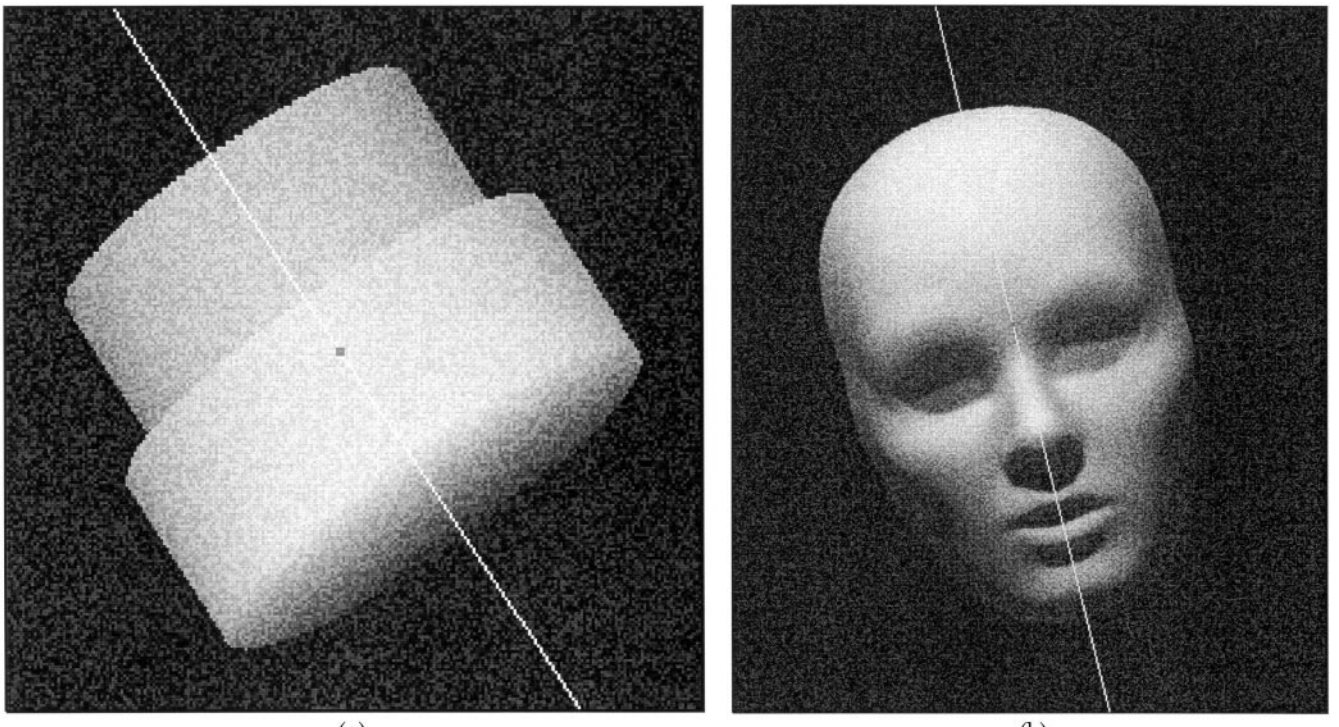


Figure 10. Symmetry detection process for a real image. (a) Original image and the symmetry axis obtained; (b) the convolution function obtained from the inverse Fourier transform. The symmetries are obtained by searching the peaks of this function.

actually reflectionally symmetric or not. These parameters or symmetry hypothesis can be verified by obtaining a symmetry measure (see Zabrodsky *et al*. [9] or O'Mara and Owens [16]). If this symmetry measure gives a low value, we say that the object is not symmetric. If an object is convex, the histogram property will not only be a necessary, but also a sufficient condition for symmetry detection. Figure 6 gives an example of a non-convex object. In this image, two symmetry axes were initially detected. One of the incorrect axes was eliminated after considering the symmetry measure.

Figure 7(a) shows an original image and the symmetry axes obtained; Figure 7(b) is the gradient orientation image normalized from 0−360 to 0−255 for display purposes; Figure 7(c) gives the histogram of the gradient orientation image (b); and Figure 7(d) shows the convolution function of the histogram obtained by using the Fourier method. The two high peaks correspond to the two symmetry axes. Figure 8 shows the results of the symmetry detection algorithm on several synthetic images. Figure 9 gives another example of symmetry detection. Figure 9(b) is the gradient orientation histogram, and Figure 9(c) is the convolution function. There is only one peak within 90% of the maximum value. Figure 10 illustrates the steps for obtaining the symmetry axes for a real image, an image of Jupiter. As shown in Figure 10(b), there are two peaks that are within 90% of the maximal value. The symmetry evaluation step indicates that the symmetry measures (using the measure described in [16]) for these two peak positions are very similar. Although our human perception may prefer to choose the axis running from the top of the image to the bottom of the image [see Figure 10(a)], the algorithm actually gives a slightly stronger symmetry measure for the other axis. After careful examination about the image we find that the change of intensity in the image around the left boundary of Jupiter is smaller than the change around the right boundary of Jupiter. It is this difference in the change of intensity that makes the algorithm pick up two symmetry axes. Figure 11 shows the results of the algorithm on two noise added synthetic images. The symmetry axes are very close to the ones obtained in the noise free images. Figure 12 gives the results of the algorithm on some real images.

The typical CPU running time for a 256 × 256 image



(a) (b)

Figure 11. Symmetry detection for two noise added synthetic images. (a) 25% noise added to the image of Figure 8(a); and (b) 25% noise added to the image of Figure 8(f).

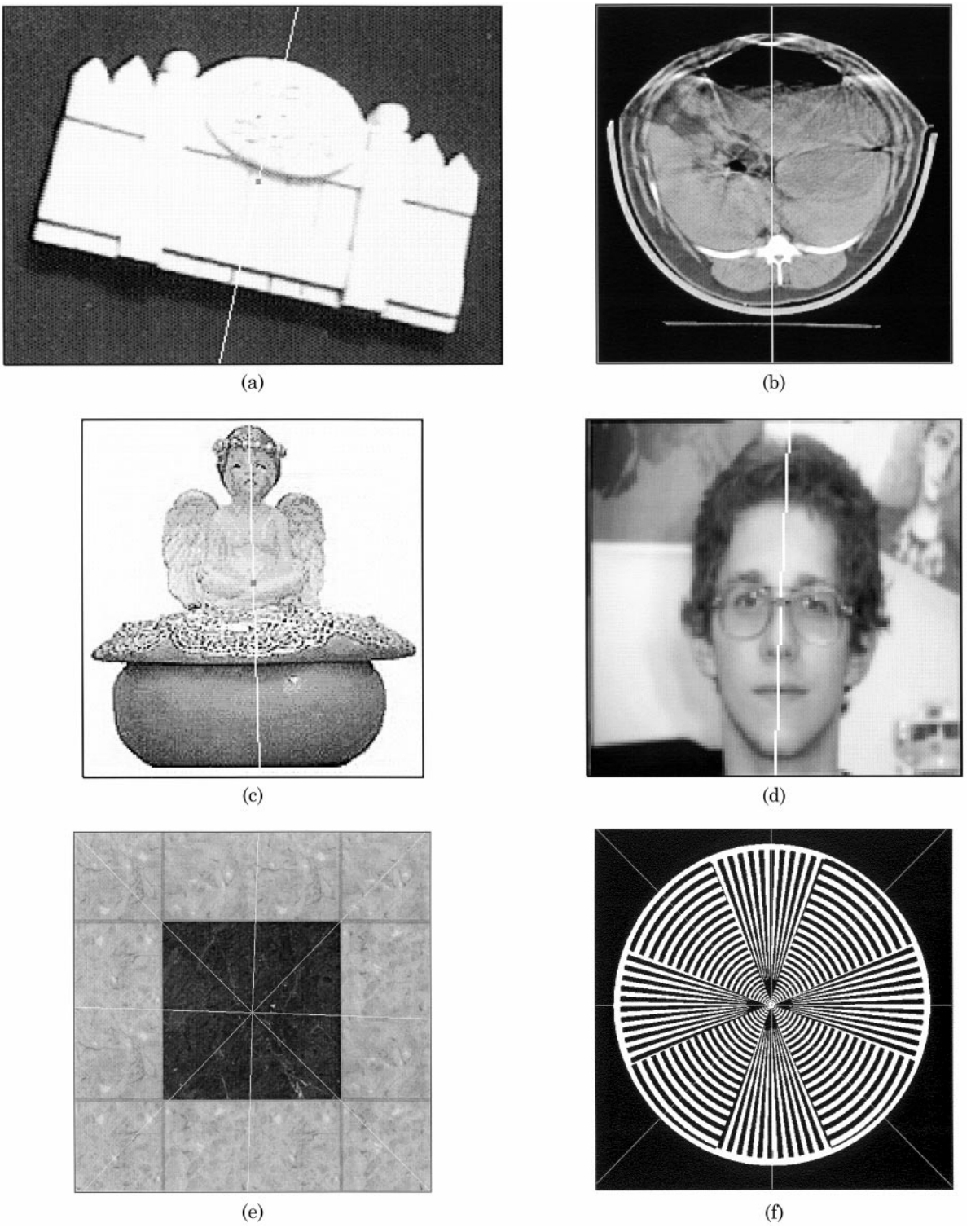


Figure 12. Symmetry detection results for several real images. (a) Welcome gate; (b) slice of CT x-ray image; (c) ceramic; (d) human face; (e) square pattern; and (f) fluctuating crosses.

is less than 0.1 s on a Sun Sparc10. The time spent on the calculation of symmetry orientation from the obtained histogram is about 2.6 ms, while the method used in [17] takes about 72.7 ms. The code has not been optimized for speed-up of processing.

Two other related methods for symmetry detection were implemented and the results were compared with our method. The reason for choosing these methods for comparison is that they mainly detect symmetry from intensity images rather than using contour points or isolated point sets. Table 1 shows some of the results obtained. The methods by Zielke *et al*. [12, 13] were not compared here, as they only looked at vertical or near vertical symmetry axes. Masuda *et al*. [14] used directional pattern matching of edge feature to detect rotational and reflectional symmetry. The main disadvantages of this method were its computational cost and memory requirements.

It is difficult to compare different methods in all aspects, as each method has its own emphases. The implementation details may be different for different people as to data I/O, memory handling, etc. The following comparison is made on the basis that the symmetry axes pass through the center of mass of the objects, and the algorithms work on the gray level images. All the methods listed in Table 1 are theoretically simple and fast for gray level images compared with other methods mentioned in the Introduction section.

The method by O'Mara and Owens [16] is fast and simple to implement, but the accuracy of the algorithm depends on the results of the principle axes, which are sensitive to imperfect symmetries. The method only selects the dominant symmetry. The method by Sun [17] is also fast and simple to implement. It only detects the most likely symmetry axis. The method proposed in this paper is fast to compute and simple to implement. It is also able to detect multiple symmetry axes.

Table 1. Comparison of three methods for reflectional symmetry detection

	CPU time (s)	Multiple axes
O'Mara & Owens (1996)	~0.05	No
Sun (1995)	~0.15	No
Our algorithm	~0.10	Yes

Conclusions

Based on our observation about the relationship between object symmetry and the gradient orientation histogram, a simple and fast reflectional symmetry detection algorithm has been developed which employs only the original gray scale image and its gradient orientation information. Fast Fourier transform has been used to obtain the convolution function from the gradient orientation histogram of the input image. The peaks in the obtained convolution function correspond to the orientations of symmetry axes. The positions of these peaks can be refined by fitting a parabola function in the neighbourhoods of these peaks. The refined positions are obtained analytically. This method can detect multiple symmetry axes of an object. The CPU time for a 256×256 image takes less than 0.1 s on a Sun Sparc10. The information about the symmetry axis can be used for many applications (such as robot operations, further image segmentation, etc).

Work has only been performed on a single object in an image. The symmetry axes for separated multiple objects can be obtained after the image segmentation of these objects. For every single object, the algorithm described in this paper is applied to obtain the symmetry axes.

Acknowledgements

We thank the reviewers for their valuable comments. The authors would also like to thank Professor N. Wade for his kind permission to use one illustration from his book: *Visual Allusions: Pictures of Perception*, 1990.

References

1. Marr, D. (1982) *Vision*. San Francisco, CA: Freeman.
2. Burton, F. W., Kollias, J. G. & Alexandridis, N. A. (1984) An implementation of the exponential pyramid data structure with application to determination of symmetries in pictures. *Computer Vision, Graphics, and Image Processing*, **25**: 218–225.
3. Wolter, J. D., Woo, T. C. & Volz, R. A. (1985) Optimal algorithms for symmetry detection in two and three dimensions. *The Visual Computer*, **1**: 37–48.
4. Atallah, M. J. (1985) On symmetry detection. *IEEE Transactions on Computers*, **C-34**: 663–666.
5. Davis, L. S. (1977) Understanding shape: II. Symmetry. *IEEE Transactions on Systems, Man and Cybernetics*, pp. 204–212.
6. Highnam, P. T. (1986) Optimal algorithms for finding the

symmetries of a planar point set. *Information Processing Letters*, **22**: 219–222.
7. Marola, G. (1989) On the detection of the axes of symmetry of symmetric and almost symmetric planar images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11**: 104–108.
8. Marola, G. (1989) Using symmetry for detecting and locating objects in a picture. *Computer Vision, Graphics, and Image Processing*, **46**: 179–195.
9. Zabrodsky, H., Peleg, S. & Avnir, D. (1992) A measure of symmetry based on shape similarity. In: *Proceedings of Computer Vision and Pattern Recognition*, pp. 703–706.
10. Zabrodsky, H., Peleg, S. & Avnir, D. (1992) Hierarchical symmetry. In: *Proceedings of International Conference on Pattern Recognition*, **III**: 9–11.
11. Parry-Barwick, S. & Bowyer, A. (1993) Symmetry analysis and geometric modelling. In: Fung, K. K. & Ginige, A. (eds.) *Digital Image Computing: Techniques and Applications*, **I**: 39–46. Australian Pattern Recognition Society.
12. Zielke, T. Brauckmann, M. & von Seelen, W. (1992) CARTRACK: computer vision-based car-following. In: *Workshop on Applications of Computer Vision*, pp. 156–163.
13. Zielke, T., Brauckmann, M. & von Seelen, W. (1993) Intensity and edge-based symmetry detection with an application to car-following. *Computer Vision, Graphics, and Image Processing: Image Understanding*, **58**: 177–190.
14. Masuda, T., Yamamoto, K. & Yamada, H. (1993) Detection of partial symmetry using correlation with rotated-reflected images. *Pattern Recognition*, **26**: 1245–1253.
15. Bowns, L. & Morgan, M. J. (1993) Facial features and axis of symmetry extracted using natural orientation information. *Biological Cybernetics*, **70**: 137–144.
16. O'Mara, D. & Owens, R. (1996) Measuring symmetry in digital images. In: *Proceedings of TENCON '96–IEEE Region Ten Conference: Digital Signal Processing Applications*, volume I, pages 151–156.
17. Sun, C. (1995) Symmetry detection using gradient information. *Pattern Recognition Letters*, **16**: 987–996.
18. Bracho, R. & Sanderson, A. C. (1985) Segmentation of images based on intensity gradient information. In: *Proceedings of Computer Vision and Pattern Recognition*, San Francisco, CA, pp. 341–347.
19. Danielsson, P. E. & Seger, O. (1990) Generalized and separable Sobel operators. In: Freeman, H. (ed.) *Machine Vision for Three-Dimensional Scenes*, pp. 347–379. Academic Press.