Chapter 3

Research Designs and Methods

This chapter describes the research designs and methods of the proposed technique. Section 3.1 introduces the model of a novel boundary detection technique. Section 3.2 describes the average edge vector field model. Section 3.3 describes the edge map. Section 3.4 describes the technique for finding the initial positions of the edge following technique. Section 3.5 describes the technique for finding the boundaries of objects in noisy images.

3.1 Model of the Novel Boundary Detection Technique

The edge following technique can detect the boundaries of objects in noisy images using information from the vector image model and the edge map. It is exploited to develop a new boundary extraction technique. The vector image model is derived by averaging edge vector fields in which both direction and magnitude are taken into account. The edge vector field is the edge vectors from current loops encompassing the objects. The edge map is derived from texture features and Canny edge detection. The vector image model and the edge map are used to select the best edges from the edge following technique. The model of the proposed boundary detection techniques can be summarized as shown in Figure 3.1.

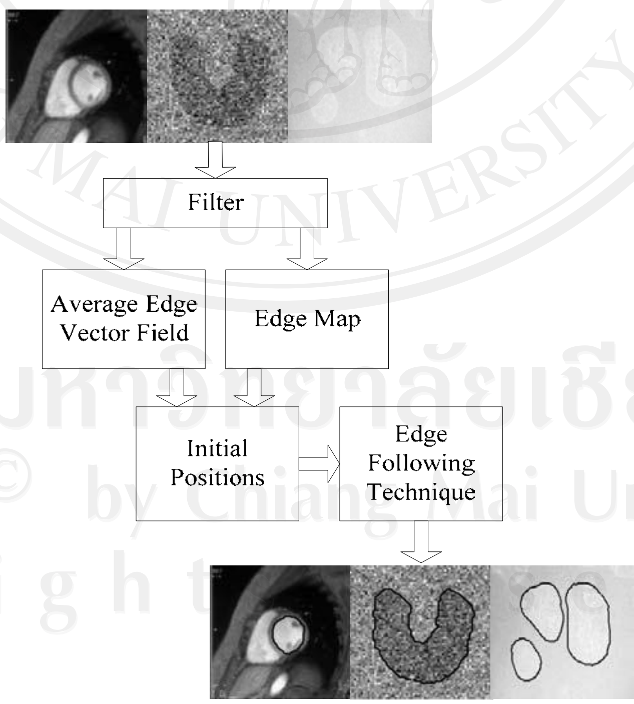


Figure 3.1 Model of the proposed boundary detection technique.

Input images are noisy images such as medical images. An input image in noisy image is filtered by using median filter. The edge vector fields and edge map are calculated. After the initial edge point is assigned, the edge following can detect the correct boundaries of objects in noisy image.

3.2 Average Edge Vector Field Model

We exploit the edge vector field [35],[36] to devise a new boundary extraction algorithm. Given an image $f(x,y)$, the edge vector field is calculated according to the following equations:

$$\vec{e}(i,j) = \frac{1}{k}\left(M_x(i,j)\vec{i} + M_y(i,j)\vec{j}\right), \tag{3.1}$$

or

$$\vec{e}(i,j) \approx \frac{1}{k}\left(\frac{\partial f(x,y)}{\partial y}\vec{i} - \frac{\partial f(x,y)}{\partial x}\vec{j}\right), \tag{3.2}$$

where

$$k = \max_{i,j}\left(\sqrt{M_x(i,j)^2 + M_y(i,j)^2}\right). \tag{3.3}$$

Each component is the convolution between the image and the corresponding difference mask, i.e.,

$$M_x(i,j) = -G_y * f(x,y) \approx \frac{\partial f(x,y)}{\partial y}, \tag{3.4}$$

$$M_y(i,j) = G_x * f(x,y) \approx -\frac{\partial f(x,y)}{\partial x}, \tag{3.5}$$

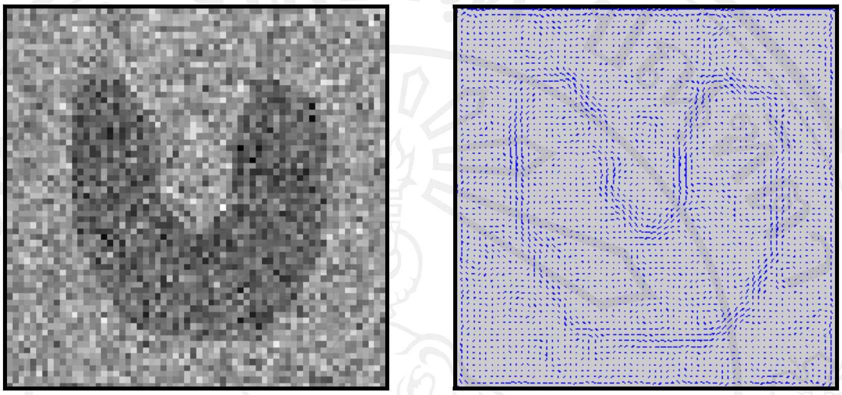
where difference masks G_x and G_y are the difference masks of the Gaussian weighted image moment vector operator in *x* and *y* directions:

$$G_x(x,y) = \frac{1}{\sqrt{2\pi\sigma}}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right).\left(\frac{x}{\sqrt{x^2+y^2}}\right), \tag{3.6}$$

and

$$G_y(x,y) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)\cdot\left(\frac{y}{\sqrt{x^2+y^2}}\right). \qquad (3.7)$$

An example of the edge vector field is shown in Figure 3.2.



(a) (b)

Figure 3.2 (a) Original image, (b) Edge vector field.

Edge vectors of an image indicate the magnitudes and directions of edges which are in the form of vector streams flowing around an object. The vectors from unclear images such as medical images and noisy images may distribute randomly in magnitude and direction. To solve this problem, we apply a local averaging operation where the value of each vector is replaced by the average of all the values in the local neighborhood. They are calculated according to the following equations:

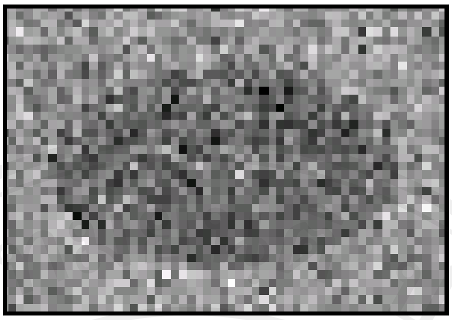
$$M(i,j) = \frac{1}{N_r}\sum_{(x,y)\in N}\sqrt{M_x(i,j)^2 + M_y(i,j)^2} \qquad (3.8)$$

and

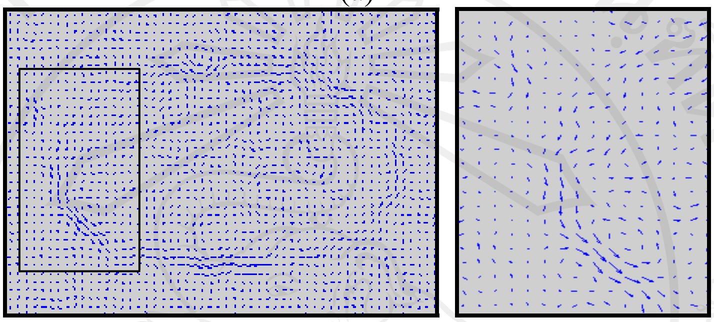
$$D(i,j) = \frac{1}{N_r}\sum_{(x,y)\in N}\tan^{-1}\left(\frac{M_y(i,j)}{M_x(i,j)}\right), \qquad (3.9)$$

where N_r is the total number of pixels in the neighborhood N.

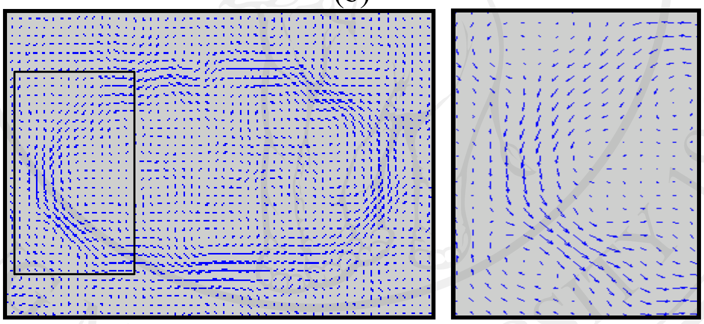
An example of the averaged edge vector field is shown in Figure 3.3.



(a)



(b)



(c)

Figure 3.3 Vector fields of unclear image (a) Original image,
(b) Results from edge vector field and zoomed-in image,
(c) Results from averaging of edge vector field and zoomed-in image.

This idea is exploited for the boundary extraction algorithm of object in noisy images. Figures 3.3(b) and 3.3(c) show that our proposed edge vector field is more descriptive along the object edges than the original edge vector field.

3.3 Edge Map

Edge map is edges of objects in an image derived from Laws' texture and Canny edge detection. It is comparable to map of edge in the image. It gives important information of the boundaries of objects in the image which is exploited in a decision for edge following.

The texture feature images of Laws' texture [37],[38] are computed by convolving an input image with each of the masks. A two-dimensional convolution mask used for texture discrimination is generated from a one-dimensional convolution mask, i.e. L5L5, as shown in Figure 3.4. We can obtain the output image by convolving the input image with the texture mask, i.e.,

$$t(i,j) = l(i,j) * f(i,j)\,, \tag{3.10}$$

$$t(i,j) = \sum_{m=-2}^{2}\sum_{n=-2}^{2} l(m,n) f(i+m, j+n)\,, \tag{3.11}$$

where $l(i,j)$ is the two-dimentional L5L5 mask, $f(i,j)$ is the input image and $t(i,j)$ is the output image.

$$\underset{L5}{\begin{bmatrix}1\\4\\6\\4\\1\end{bmatrix}} \underset{L5}{\begin{bmatrix}1 & 4 & 6 & 4 & 1\end{bmatrix}} = \underset{L5L5}{\begin{bmatrix}1 & 4 & 6 & 4 & 1\\4 & 16 & 24 & 16 & 4\\6 & 24 & 36 & 24 & 6\\4 & 16 & 24 & 16 & 4\\1 & 4 & 6 & 4 & 1\end{bmatrix}}$$

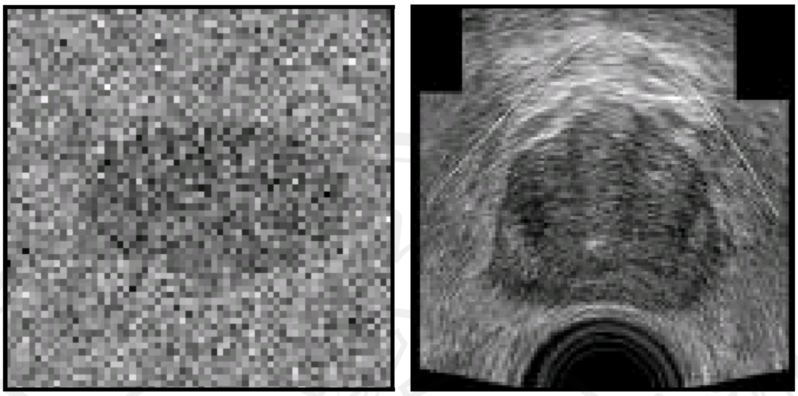
Figure 3.4 Laws' mask derived from L5L5.

The first step of Canny edge detection is to convolve the output image which obtained from the aforementioned Laws' texture $t(i,j)$ with a Gaussian filter $g(i,j:\sigma)$, i.e.,

$$S(i,j) = g(i,j:\sigma) * t(i,j)\,, \tag{3.12}$$

where σ is the spread of the Gaussian and controls the degree of smoothing, $t(i,j)$ is the output image from Laws' texture process and $g(i,j:\sigma)$ is the Gaussian filter. The following steps of the Canny edge detection follow what are already described in Chapter 2.

Edge map shows some important information of edge. This idea is exploited for extracting the correct boundaries of objects in an unclear images. An example of the edge map is shown in Figure 3.5.



(a)

(b)

Figure 3.5 (a) Original images, (b) Corresponding edge maps derived from Laws' texture (L5L5) and Canny edge detection.

3.4 Initial Positions

In this section we present a technique for determining good initial positions of edge following that can be used for the correct boundary detection. This step is to find an automatic initialization technique. Initial position problem is very important in the classical contour models. Snakes models may converge to a wrong boundary if the initial position is not close enough to the desired boundary. Finding the initial position of the classical contour models is still difficult and time consuming. In the proposed technique, it is easy to find the initial positions of edge following. The initial position of edge following is determined by the following steps.

The first step is to calculate the average magnitude ($M(i, j)$) which is derived from equation (3.8). The first basic idea of determining the initial positions is the magnitude. The high magnitude will be the strong edges on the image which is essential information for selecting the initial position of edge following.

The second step is to calculate the density of edge length for each pixel from the edge map. Edge map ($E(i, j)$), as a binary image, is obtained by Laws' texture and Canny edge detection. The idea of density is to obtain measurement of the edge length. The density of edge length ($L(i, j)$) in each pixel can be calculated from

$$L(i,j) = \frac{C(i,j)}{Max(C(i,j))} \tag{3.13}$$

where $C(i,j)$ is the number value of connected pixels on an edge in each position of pixel. An example of counting the numbers of connected pixels is shown in Figure 3.6(b). The density of the edge length from the example is shown in Figure. 3.6(c).

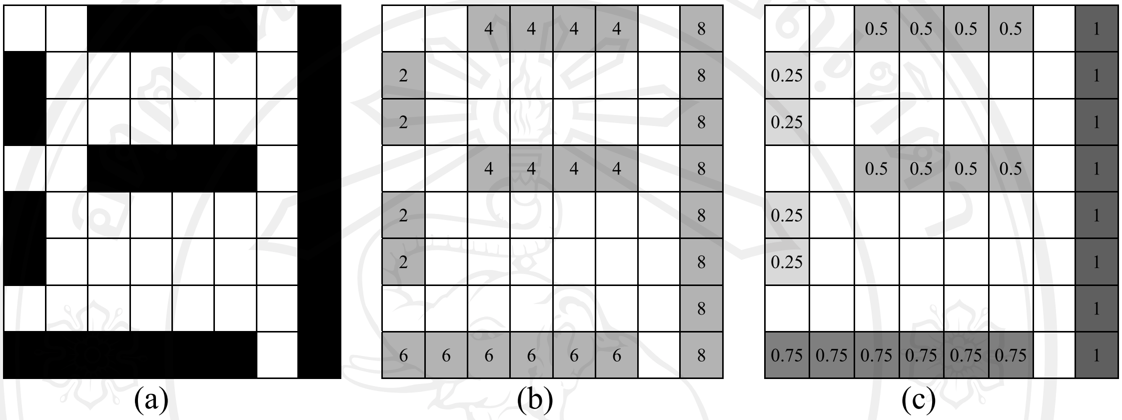


Figure 3.6 (a) Edge map ($E(i,j)$), (b) Results of counting the connected pixels ($C(i,j)$), (c) Results of calculating the density of edge length ($L(i,j)$).

The third step is to calculate the initial position map $P(i,j)$ by averaging the average magnitude and the density of the edge length, i.e.,

$$P(i,j) = \frac{1}{2}(M(i,j) + L(i,j)). \tag{3.14}$$

The final step is the thresholding of the initial position map. We have to threshold the initial positions map in order to detect the good initial positions of edge following. If $P(i,j) > T_{\max}$, then $P(i,j)$ is an initial position of edge following. $T_{\max}$ is the threshold value which is set by a user. The initial positions obtained by our method are positions that close to the edges of interested areas. Examples of the initial positions derived from our method are shown in Figure 3.7 and Figure 3.8.

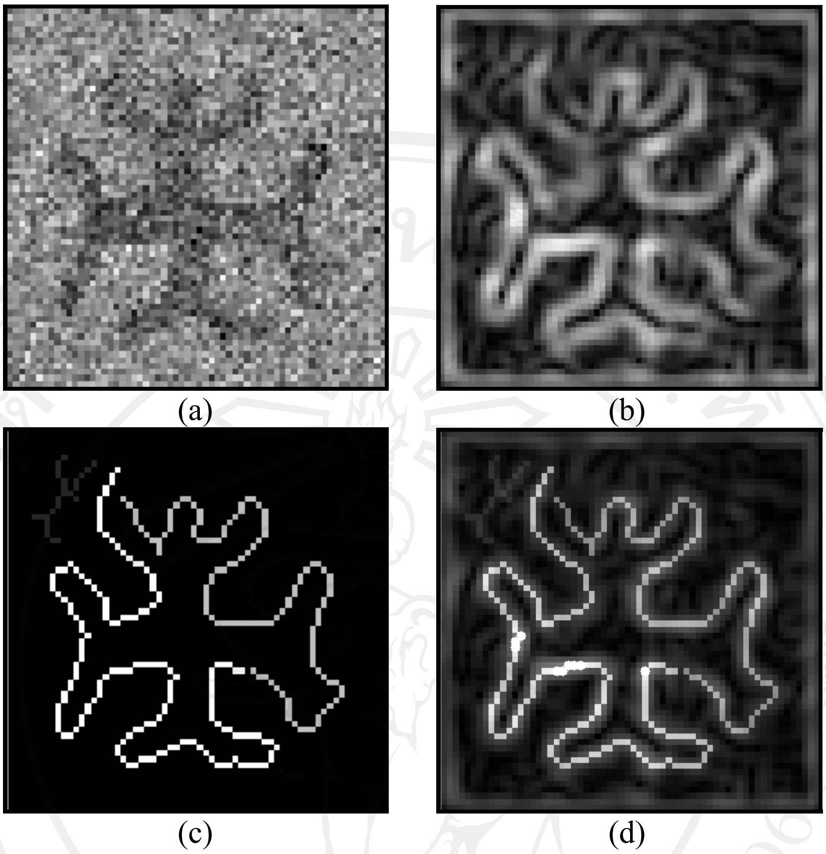


Figure 3.7 (a) Synthetic noisy image, (b) The average magnitude ($M(i, j)$), (c) The density of length edge ($L(i, j)$), (d) The initial position map ($P(i, j)$) and the initial positions of edge following derived by thresholding ($T_{max} = 0.95$).

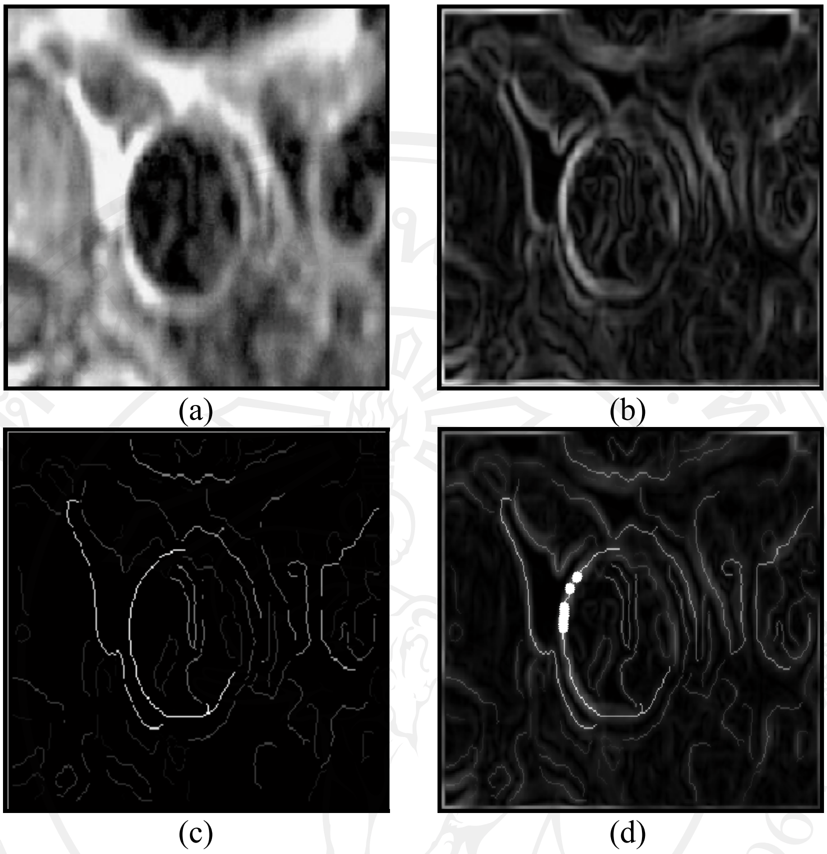
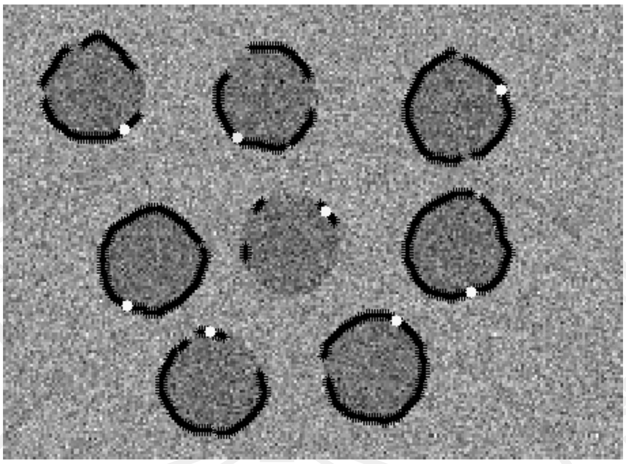


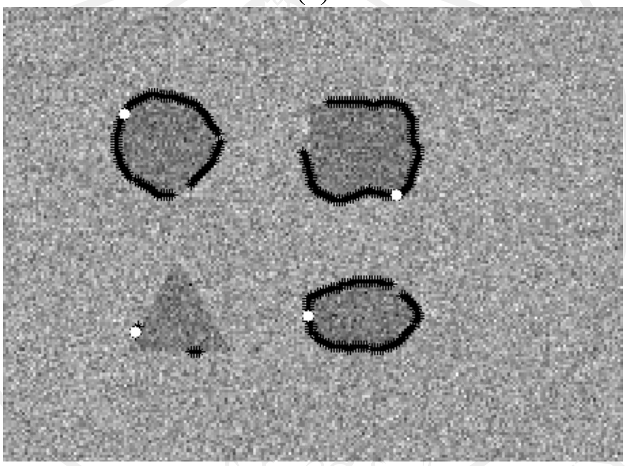
Figure 3.8 (a) Aorta in cardiovascular MR image, (b) The average magnitude ($M(i, j)$), (c) The density of length edge ($L(i, j)$), (d) The initial position map ($P(i, j)$) and the initial positions of edge following derived by thresholding ($T_{\max} = 0.95$).

The white circle points in the initial position map are the positions of initial points for our technique. All white circle points can be the initial point for detecting the boundary of object for our edge following technique. However, the maximum value of white circle points is used to be the initial point in this research.

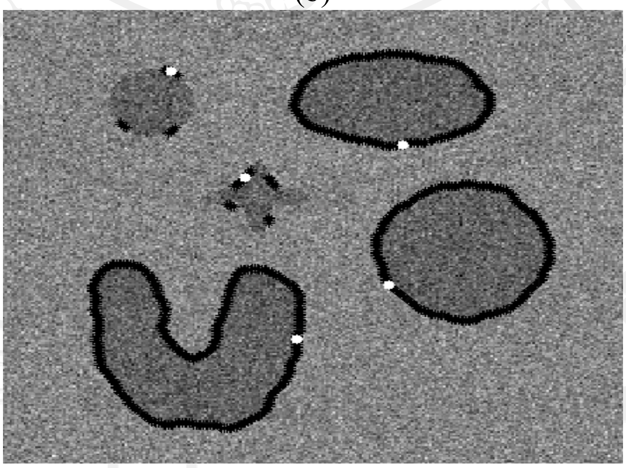
In the case of multiple objects, we found a lot of initial points in each object. So, all initial points are reduced by finding the maximum value of each object by using important information from the edge map. The initial point in each object is selected from the maximum value of $P(i,j)$ in each line of edge map. An example of the initial point derived from our method is shown in Figure 3.9. The black asterisks in the images are the initial positions of the edge following derived by thresholding. The white circles are the result from reducing the initial points. All white circles can be used as the initial points for detecting the boundaries of the objects. After determining the suitable initial position, the edge following will follow edges along the object boundary until the closed-loop contour is achieved. Some objects still have white circle points more than one. However, when the edge following technique find the next white circle points before the closed-loop contour is achieved, the next white circle points will be deleted.



(a)



(b)



(c)

Figure 3.9 Synthetic noisy images in the case of multiple objects and the initial positions of edge following derived by thresholding ($T_{\max} = 0.50$) and result from reducing the initial points (white circle points).

3.5 Edge Following Technique

The edge following technique is performed to find the boundary of an object. Most edge following algorithms take into account the edge magnitude as primary information for edge following [33],[34]. However, the edge magnitude information is not enough for searching the correct boundary of an object in noisy images because it can be very weak in some contour areas. This is exactly the reason why the edge following techniques failed to extract the correct boundary of object in noisy images. To remedy the problem, we propose a new edge following technique by using information from the average edge vector field and edge map. It gives more information for searching the boundary of object and increases the probability of searching the correct boundary. From the average edge vector field, the magnitude and direction give information of the boundary of object which flows around an object. From the edge map, it gives information of edge which may be boundary of object. At the position (i, j) of an image, the successive positions of the edges are calculated using the following equation:

$$\mathbf{L}_{ij} = \alpha\mathbf{M}_{ij} + \beta\mathbf{D}_{ij} + \varepsilon\mathbf{E}_{ij}, \tag{3.15}$$

where α, β, and ε are the weight parameters that control the position of edge following. The total value of all weight parameters are set to 1. i and j are the row and column of a pixel in the image, respectively.

At the position (i, j) of an image, the matrices $\mathbf{M}_{ij}, \mathbf{D}_{ij}$ and $\mathbf{E}_{ij}$ are calculated as follows:

$$\mathbf{M}_{ij}(r,c) = \frac{M(i+r-1, j+c-1)}{\max M(i,j)}, \quad 0 \le r,c \le 2 \tag{3.16}$$

$$\mathbf{D}_{ij}(r,c) = 1 - \frac{\left|D(i,j) - D(i+r-1, j+c-1)\right|}{\pi}, \quad 0 \le r,c \le 2 \tag{3.17}$$

$$\mathbf{E}_{ij}(r,c) = E(i+r-1, j+c-1), \quad 0 \le r,c \le 2 \tag{3.18}$$

where

$M(i,j)$ is the average magnitude of edge vector fields as shown in equation (3.8),

$D(i,j)$ is the average direction of edge vector fields as shown in equation (3.9),

$E(i,j)$ is the edge map from Laws' texture and Canny edge detection.

It should be noted that the range of $M(i,j)$, $D(i,j)$, and $E(i,j)$ are between 0 to 1.

Finally, at the position (i,j), the most likely direction linked to the next edge pixel can be calculated by

$$D_{ij,opt} = \arg\max_{k} \sum_{r=0}^{2}\sum_{c=0}^{2} \mathbf{L}_{ij}(r,c)\mathbf{C}_{k}(r,c), \qquad (3.19)$$

where $k = 1,2,\ldots,8$, denotes the 8 directions as indicated by the arrows at the center of each mask shown in Figure 3.10. $\mathbf{C}_k$ is a constraint of edge following shown in Figure 3.10. The constraint mask is selected by considering the direction of the vector model at a position in $D(i,j)$. The mask which has a similarity in direction of vector is selected to suit the chosen constraint of edge following. The value of each element in each mask dictates the corresponding direction. The constraint may be determined by the additional constraints set by the user and a higher-level process. The edge following is started from initial positions to its corresponding end position.

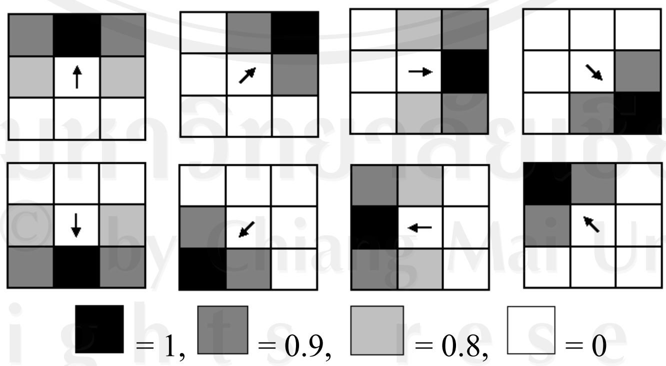


Figure 3.10 Edge masks used for detecting of image edges (normal direct constraint).

The edge following technique is performed to find the closed-loop contours. After determining suitable initial position, the edge following can detect the boundary of object in images until the closed-loop contour is found.