CHAPTER 3

Related Theories

This chapter provides an overview of related theories used in our research.

3.1 Digital image conversion [17, 18]

In digital image processing, an image can be defined as a two-dimensional function $f(x, y)$, where x and y are spatial coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the intensity of the image at that point. The gray level is a term frequently used to refer to the intensity of monochrome images. A combination of individual 2-D images does create a color image. For example, for the RGB color system, its color image contains red, green, and blue, as the individual components of the image. 0s and 1s are logical arrays of a binary image. We implement the following theories in our research.

Color conversion to gray scale was frequently used in this study. The conversion from RGB to gray image (I) can be denoted by

$$I(x, y) = 0.2989R(x, y) + 0.5870G(x, y) + 0.1140B(x, y) \tag{3.1}$$

3.2 Morphological Image Processing [17]

Morphology is the word generally denotes a branch of biology that concerns the structure and form of animals and plants. However, we use the word mathematical morphology as a tool for extracting image components, in representation and description of region shape such as boundaries, skeletons, and the convex hull.

Morphological Image Processing used in our research consists of dilation, erosion, opening, closing and labeling. The principles of them are as follows.

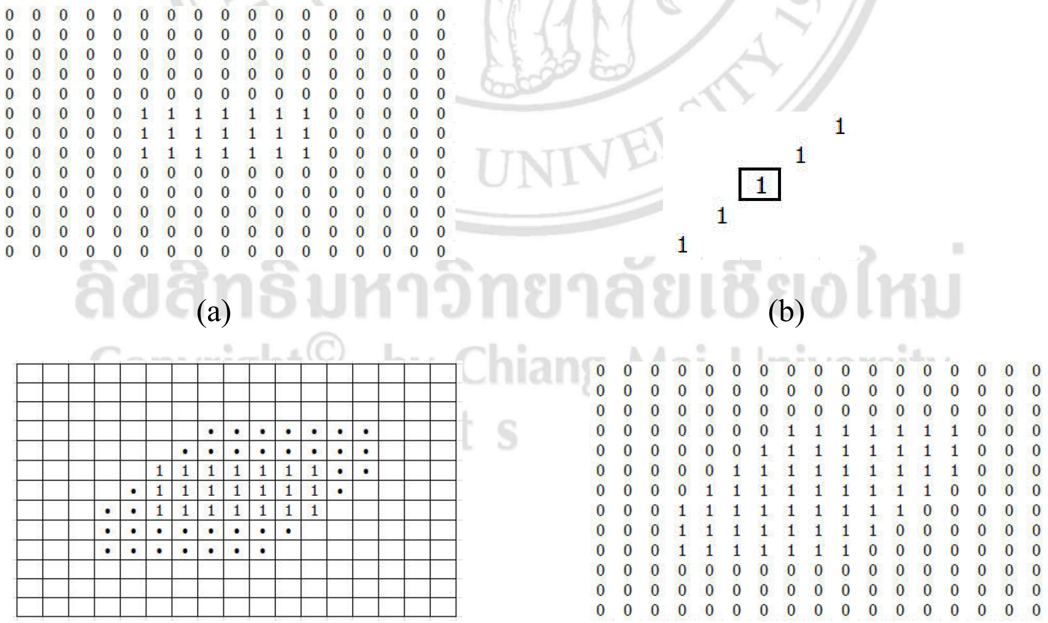
3.2.1 Dilation

Dilation is an algorithm that thickens objects in a binary image. The thickening of the object is controlled by a shape called a structuring element. Figure 3.1 shows how dilation works. Figure 3.1(a) presents a simple binary image consisting of a rectangular object. Figure 3.1(b) is a structuring element and in this case is a five-pixel-long diagonal line. Figure 3.1(c) shows dilation as a process that translates the origin of the structuring element throughout the domain of the image and find the overlaps with 1-valued pixels. The output image in figure 3.1(d) is the result of placing the structural element by the center of its overlapped on each 1 value pixels in the input image. The dilation of A by B, denoted $A \oplus B$, is defined as

$$A \oplus B = \left\{ z \middle| \left(\hat{B}\right)_z \cap A \neq \varnothing \right\} \tag{3.2}$$

where A is the binary image.

B is the structuring element.



(a) (b)

(c) (d)

Figure 3.1 Dilation methodology, (a) rectangular object in the original image, (b) structuring element, (c) structuring element moved and translated to all 1-value pixel locations on the image, (d) output image.

3.2.2 Erosion

Erosion shrinks objects in a binary image. As in dilation, shrinking is controlled by a structuring element. Figure 3.2 shows the erosion process. Figure 3.2(a) is similar to figure 3.1(a). Figure 3.2(b) is the structuring element which is a short vertical line. Figure 3.2(c) show erosion as a process for translating the structuring element of the image domain and find out where it fits with the image foreground. The output image in Figure 3.2(d) has the value of 1 at all location of the origin of the structuring element, such that only 1-valued pixels of the input image is overlapped by the element. The erosion of *A* by *B*, denoted *A* Θ *B*, is defined as

$$A \,\Theta\, B = \left\{ z \left| \left(\hat{B}\right)_z \cap A^c \neq \varnothing \right. \right\} \tag{3.3}$$

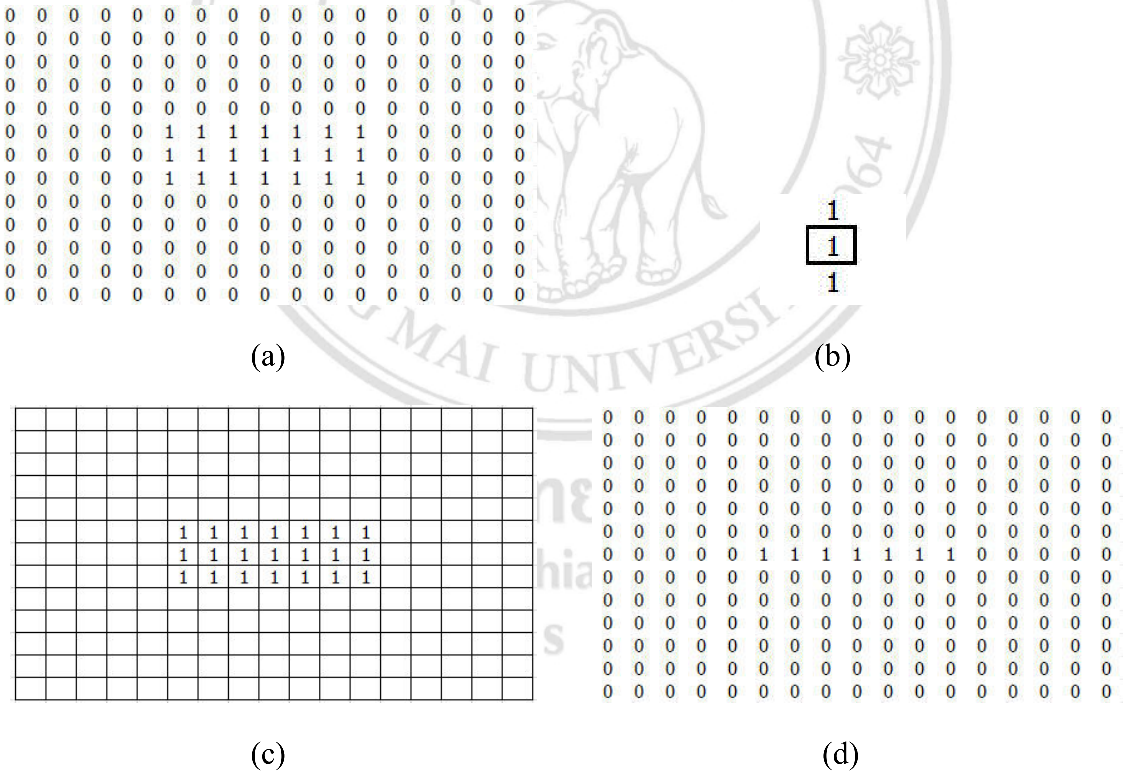


Figure 3.2 Erosion methodology, (a) original image with rectangular object, (b) structuring element, (c) structuring element translated to several locations on the image, (d) output image.

3.2.3 Opening

The morphological opening of A by B, denoted $A \circ B$, is commonly erosion of A by B and then dilation of the result by B

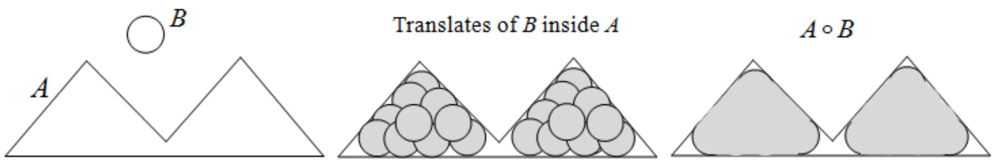
$$A \circ B = (A \ominus B) \oplus B \tag{3.4}$$

An alternative mathematical equation of opening is

$$A \circ B = \cup \left\{ (B)_z \,\middle|\, (B)_z \subseteq A \right\} \tag{3.5}$$

where $\cup\{\cdot\}$ is the union of all sets inside the braces, and $(B)_z \subseteq A$ means that $(B)_z$ is a subset of A.

Figure 3.3 illustrates the result of opening operation. Figure 3.3(a) illustrates a set A and structuring element B in a disk-like shape. Figure 3.3(b) illustrates the amount of translation of B that fits completely within A. The union of all translations is the gray region in figure 3.3(c). The white regions are not part of the opening. Morphological opening is generally used for breaking thin connections, smoothening object contours, and removing thin protrusions.



(a) (b) (c)

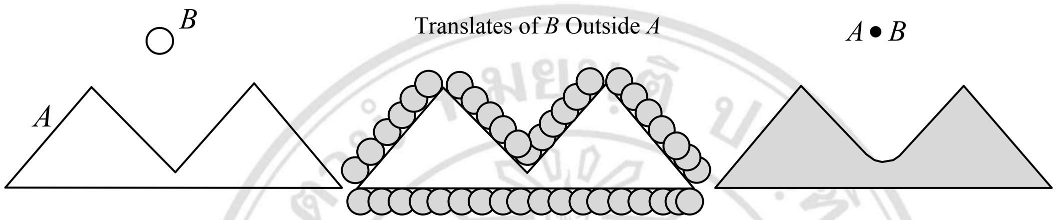
Figure 3.3 Opening methodology, (a) set A and disk structuring element B, (b) translation of B that fits completely within set A, (c) the complete opening (gray color).

3.2.4 Closing

The morphological closing of A by B, denoted $A \bullet B$ which is a dilation followed by an erosion can be written in the form

$$A \bullet B = (A \oplus B) \ominus B \tag{3.6}$$

where $A \bullet B$ is the complement of the union of all translation of B that does not overlap A. Figure 3.4(b) shows the translation of B that does not overlap A. By taking the union complementation of all translation, we get the gray region if figure 3.4(c) is a complete closing.



(a) (b) (c)

Figure 3.4 Closing methodology, (a) set A and structuring element B, (b) translation of B outside A, (c) the complete closing (gray color).

Similar to opening, closing morphology tends to smooth the contours of objects. Opposite to opening, it fills long thin gulfs, commonly joins narrow breaks together, and fills holes smaller than the structuring element.

3.2.5 Labeling Connected Components

The connected component was called a path, and the definition of a path depends on adjacency. So that the nature of a connected component depends on which form of adjacency is chosen (commonly with 4- and 8-adjacency). Figure 3.5(a) illustrates a binary image with four 4-connected components. Figure 3.5(b) illustrates that choosing 8-adjacency can reduces the number of connected components to two.

1	1	1	0	0	0	0	0
1	1	1	0	1	1	0	0
1	1	1	0	1	1	0	0
1	1	1	0	0	0	1	0
1	1	1	0	0	0	1	0
1	1	1	0	0	0	1	0
1	1	1	0	0	1	0	0
1	1	1	0	0	0	0	0

(a)

1	1	1	0	0	0	0	0
1	1	1	0	1	1	0	0
1	1	1	0	1	1	0	0
1	1	1	0	0	0	1	0
1	1	1	0	0	0	1	0
1	1	1	0	0	0	1	0
1	1	1	0	0	1	0	0
1	1	1	0	0	0	0	0

(b)

1	1	1	0	0	0	0	0
1	1	1	0	2	2	0	0
1	1	1	0	2	2	0	0
1	1	1	0	0	0	4	0
1	1	1	0	0	0	4	0
1	1	1	0	0	0	4	0
1	1	1	0	0	3	0	0
1	1	1	0	0	0	0	0

(c)

1	1	1	0	0	0	0	0
1	1	1	0	2	2	0	0
1	1	1	0	2	2	0	0
1	1	1	0	0	0	2	0
1	1	1	0	0	0	2	0
1	1	1	0	0	0	2	0
1	1	1	0	0	2	0	0
1	1	1	0	0	0	0	0

(d)

Figure 3.5 Connected components, (a) four objects of 4- connected components, (b) two objects of 8-connected components, (c) label matrix result by using 4-connectivity, (d) label matrix result by using 8-connectivity.

3.3 Thresholding [17]

In the case of the intensity histogram shown in figure 3.6 is related to an image, $f(x, y)$, consisted of light objects on a dark background, such that object and background pixels have intensity levels that can be separated into two dominant modes. The objects can be extracted from the background by selecting a threshold T that separates these modes. Then any point (x, y) for which $f(x, y) \geq T$ is called an object point; otherwise, the point is called a background point. In another way, the thresholding image $g(x, y)$ is defined as

$$g(x, y) = \begin{cases} 1 \text{ if} f(x, y) \geq T \\ 0 \text{ if} f(x, y) < T \end{cases} \tag{3.7}$$

Pixels labeled 1 are the objects, whereas pixels labeled 0 are the background.

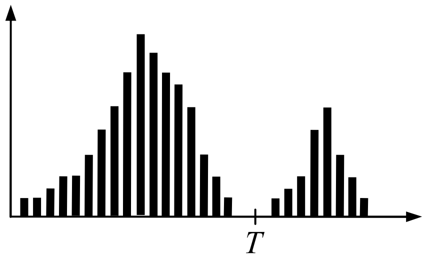


Figure 3.6 Choosing a threshold (T) by visually analyzing a bimodal histogram.

One technique to select a threshold is visual inspection of the image histogram. There are two distinct modes that can be observed by the histogram in figure 3.6. In such case, it is easy to choose a threshold *T*. Another method to select *T* is by trial and error which randomly chooses different thresholds until a good result is found as judged by the observer.

3.4 Image Rotation

In case that an image *f* is defined over a (w,z) coordinate system, undergoes geometric distortion to produce an image *g* defined over an (x,y) coordinate system. This transformation (of the coordinates) can be shown as

$$(x,y) = T\{(w,z)\} \tag{3.8}$$

where *T* is an affine matrix.

For image rotation, the affine matrix is

$$T = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.9}$$

Then, the coordinate equation is

$$\begin{aligned} x &= w\cos\theta - z\sin\theta \\ y &= w\sin\theta + z\cos\theta \end{aligned} \tag{3.10}$$

3.5 Optical Flow [19], [20]

This methodology is based on a differential technique computed by using a gradient constraint (brightness constancy) with global smoothness to obtain an estimated velocity field (Horn and Schunck 1981).

There are two processes for the implementation of the HS methodology. The first one is an estimation of partial derivatives and the second one is a minimization of the summation of the errors by an iterative process to obtain the motion vector.

1) Estimation of Partial Derivatives

Estimate the derivatives of brightness from the discrete set of image brightness measurement available. The brightness of each pixel (E_x, E_y, and E_t) is constant.

Estimate E_x, E_y, and E_t using equation (3.11) at a point in the center of a cube as shown in figure 3.7.

The three partial derivatives of image brightness at the center of the cube are single estimated from the average of first different along the four parallel edges of the cube.

$$
\begin{aligned}
E_x &= \frac{1}{4}\left\{E_{i,j+1,k} - E_{i,j,k} + E_{i+1,j+1,k} - E_{i+1,j,k} + E_{i,j+1,k+1} - E_{i,j,k+1} + E_{i+1,j+1,k+1} - E_{i+1,j,k+1}\right\} \\
E_y &= \frac{1}{4}\left\{E_{i+1,j,k} - E_{i,j,k} + E_{i+1,j+1,k} - E_{i,j+1,k} + E_{i+1,j,k+1} - E_{i,j,k+1} + E_{i+1,j+1,k+1} - E_{i,j+1,k+1}\right\} \\
E_t &= \frac{1}{4}\left\{E_{i,j,k+1} - E_{i,j,k} + E_{i+1,j,k+1} - E_{i+1,j,k} + E_{i,j+1,k+1} - E_{i,j+1,k} + E_{i+1,j+1,k+1} - E_{i+1,j+1,k}\right\}
\end{aligned}
\quad (3.11)
$$

Here, the unit of length is the grid spacing interval in each image frame and the unit of time is the image frame sampling period.

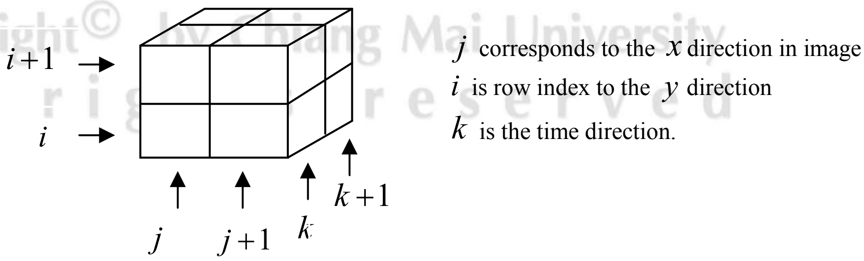


Figure 3.7 The partial derivatives of image brightness at the point (i, j)

2) Minimization

The problem then is to minimize the summation of the errors in the equation for the rate of change of image brightness,

$$\varepsilon = uE_x + vE_y + E_t = 0 \tag{3.12}$$

where u and v are the horizontal and vertical motion vector of optical flow, respectively.

One cannot expect ε to be zero. The problem is to minimize the summation of error in the equation for the rate of change of image brightness as near as 0. Therefore, the smoothness weight (α) is iteratively presented as

$$\begin{aligned} u^{n+1} &= \bar{u}^n - \frac{E_x\left[E_x\bar{u}^n + E_y\bar{v}^n + E_t\right]}{\alpha^2 + E_x^2 + E_y^2}, \\ v^{n+1} &= \bar{v}^n - \frac{E_y\left[E_x\bar{u}^n + E_y\bar{v}^n + E_t\right]}{\alpha^2 + E_x^2 + E_y^2}, \end{aligned} \tag{3.13}$$

where

$$\begin{aligned} \bar{u}_{i,j,k} &= \frac{1}{6}\left\{u_{i-1,j,k} + u_{i,j+1,k} + u_{i+1,j,k} + u_{i,j-1,k}\right\} + \\ &\quad \frac{1}{12}\left\{u_{i-1,j-1,k} + u_{i-1,j+1,k} + u_{i+1,j+1,k} + u_{i+1,j-1,k}\right\} \\ \bar{v}_{i,j,k} &= \frac{1}{6}\left\{v_{i-1,j,k} + v_{i,j+1,k} + v_{i+1,j,k} + v_{i,j-1,k}\right\} + \\ &\quad \frac{1}{12}\left\{v_{i-1,j-1,k} + v_{i-1,j+1,k} + v_{i+1,j+1,k} + v_{i+1,j-1,k}\right\} \end{aligned} \tag{3.14}$$

where $\bar{u}^k$ and $\bar{v}^k$ denote horizontal and vertical neighborhood averages (u^k and v^k) which initially are set to zero and then weighed the average of the value at neighboring points based on the kernel [1/12 1/6 1/12; 1/6 -1 1/6; 1/12 1/6 1/12].

The smoothness weight (α) plays an important role where the brightness gradient is small, for which the suitable value should be determined.

In our proposed method, we use n = 50, where n = number of iterations and $\alpha = 50$.

3.6 Optimal Thresholding [21]

Otsu's method is a well-known measure used in statistical discriminant analysis. The idea of the method is that well-threshold classes should be distinguished from the intensity values of their pixels. In terms of their intensity values, a threshold which gives the best separation between classes would be the optimum threshold.

Let $\{0,1,2,...,L-1\}$ denote L distinct intensity levels in a $M \times N$ digital image and let n_i denote the number of pixels with intensity i. Then the total number of pixels in this image is $MN = n_0 + n_1 + n_2 + ... + n_{L-1}$ and the normalized histogram has components $p_i = \dfrac{n_i}{MN}$.

Suppose we select a threshold $T(k) = k, 0 < k < L-1$ and classify the input image into two classes, C_1 and C_2, where C_1 consists of all pixels in the image with intensity values in the range $[0,k]$ and C_2 consists of all pixels in the image with intensity value in the range $[k+1, L-1]$.

From the aforementioned details, Otsu's threshold can be calculated by the following steps:

1) Compute the normalized histogram of the input image. Denote the components of the histogram by p_i, for $i = 0,1,2,...,L-1$.

2) Compute the cumulative sums, $P_1(k)$, for $k = 0,1,2,...,L-1$, by

$$P_1(k) = \sum_{i=0}^{k} p_i \tag{3.15}$$

where $P_1(k)$ = the probability that a pixel is assigned to class C_1.

3) Compute the cumulative means, $m(k)$, for $k = 0,1,2,\ldots, L-1$ by

$$m(k) = \sum_{i=0}^{k} ip_i \tag{3.16}$$

where $m(k)$ = the mean intensity value of the pixels assigned to class C_2.

4) Compute the global intensity mean, m_G by

$$m_G = \sum_{i=0}^{L-1} i p_i \quad (3.17)$$

where m_G = the average intensity of the entire image.

5) Compute the between-class variance, $\sigma_B^2(k)$, for $k = 0,1,2,\ldots, L-1$ by

$$\sigma_B^2(k) = \frac{\left[m_G P_1(k) - m(k)\right]^2}{P_1(k)\left[1 - P_1(k)\right]} \quad (3.18)$$

6) Obtain the Otsu's threshold, k^*, as the value of k for which $\sigma_B^2(k)$ is the maximum. If the maximum is not unique, obtain k^* by averaging the value of k corresponding to the various maxima detected.

3.7 Median Filtering [22]

Median filtering smoothes images by utilizing the median of the neighborhood. Tukey was the first who proposed the concept of a median filter in 1971. After that Pratt applied this method to image processing in 1978.

The algorithm is replaced the value of a pixel by the median of the intensity levels in the neighborhood of that pixel:

$$\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{median}\left\{g(s,t)\right\} \quad (3.19)$$

The value of the pixel at (x,y) is included in the computation of the median. The method of the median filter can be done following the 3 steps below:

1) Select a mask $n \times n$ (n = odd number)
2) All pixels in the neighborhood of the pixels in the original image which are identified by the mask are sorted in the ascending order.
3) The median of the sorted values is computed and is chosen as the pixel value for the processed image.

3.8 Possibilistic C-means Algorithm (PCM) [20]

The objective function of PCM can be described as follow:

Let $x = \{\bar{x}_1, \bar{x}_2, ..., \bar{x}_N\}$ be a set of data points

$v = \{\vec{v}_1, \vec{v}_2, ..., \vec{v}_C\}$ be a set of centers

$$J_m(U,V;X) = \sum_{i=1}^{C}\sum_{j=1}^{N}(\mu_{ij})^m \left(d\left(\vec{x}_j, \vec{v}_i\right)\right)^2 + \sum_{i=1}^{C}\eta_i \sum_{j=1}^{N}\left(1-\mu_{ij}\right)^m \quad (3.20)$$

$$0 < \sum_{j=1}^{N}\mu_{ij} \le N \text{ , } i = 1,2,...,C \text{ and } j = 1,2,...,N$$

where $\mathbf{U} = \left|\mu_{ij}\right|_{C\times N}$ denotes the possibilistic partition matrix.

μ_{ij} denotes the possibilistic membership of the i th cluster center to j data.

N denotes the number of data.

η_i denotes resolution or scale parameter.

m denotes the Fuzziness index $m \in [1, \infty]$.

C denotes the number of cluster centers.

$d\left(\vec{x}_j, \vec{v}_i\right)$ denotes the distance between cluster center i and j.

$\mu_{ij}, \vec{v}_i$ can be updated using the following (3.21) and (3.22):

$$\mu_{ij} = \left(1 + \left(\frac{d_{ij}^2}{\eta_i}\right)^{1/(m-1)}\right)^{-1} \quad (3.21)$$

$$\vec{v}_i = \frac{\sum_{j=1}^{N}\mu_{ij}^m x_j}{\sum_{j=1}^{N}\mu_{ij}^m}, \quad i = 1,2,...,C \text{ and } j = 1,2,...,N \quad (3.22)$$

where η_i denotes a scale parameter and it is suggested to be:

$$\eta_i = K\left(\frac{\sum_{j=1}^{N}\mu_{ij}^m \times d^2\left(\vec{x}_j, \vec{v}_i\right)}{\sum_{j=1}^{N}\mu_{ij}^m}\right) \quad (3.23)$$

where $K > 0$ (in general $K = 1$). The update stops when μ_{ij} and η_i which minimize J value are found. PCM relaxes the sum constraint column of the membership matrix in FCM so that the sum of each column of PCM partition matrix satisfies the looser constraint.

3.9 Patch-Base Possibilistic C-means

This technique was used in our heart structure segmentation method. The method of this technique started by applying 5x5 median filter to reduce the intensity inconsistency in an ultrasound image. Then PCM is used to over segment the filtered image. In our algorithm, we set the number of clusters equal to 20 and parameter m equal to 1.5. A group of pixels in each cluster is called a patch. The similar patches are then combined into two regions: heart structure and background. Heart structure (heart chamber area) is a dark area inside the heart which differs from other organs around it. The remaining area in the ultrasound image is grouped as the background area. The condition used in our algorithm to combine patches is to combine low gray level value patches together. The equation and details of calculation will be explained in detail in chapter 4, section 4.5.

The patch-based PCM methods are illustrated in the following diagram.

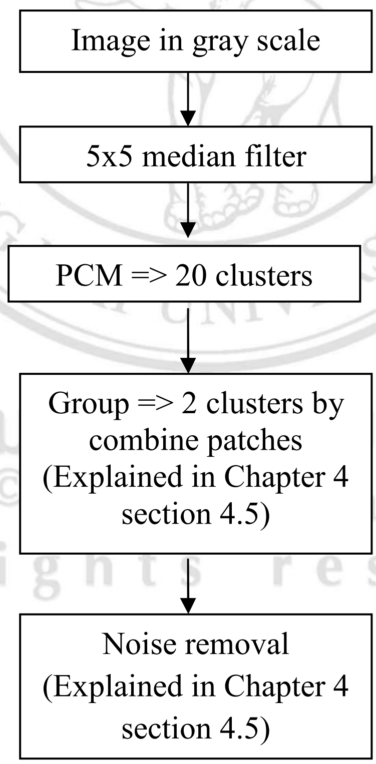


Figure 3.8 The diagram of patch-based PCM