Chapter 4
Fast Searching Algorithm in Two level Thresholding

In this chapter we will explore further the thresholding method proposed by Cheng, Chen and Li [55] which is based on the fuzzy set model of an image and the maximum entropy principle. A fast searching algorithm for the selection of the optimal threshold in two level thresholding is also presented.

4.1 Model of an Image

Let \( D \equiv \{ (i, j) : i = 0, 1, ..., M - 1; j = 0, 1, ..., N - 1 \} \), where \( M \) and \( N \) are two positive integers. Let \( G = \{ 0, 1, ..., l - 1 \} \), where \( l = 256 \) for an 8-bit image. Then a digitized image with \( l \) gray levels is considered to be a two-variable function \( I(x, y) \) with a domain \( D \) and a range \( G \), i.e., \( I(x, y) \in G, \forall (x, y) \in D \), where \( 0 \leq I(x, y) \leq l - 1 \) is the gray level value of the image at the pixel \((x, y)\). Thus, a digitized image defines a mapping \( I : D \rightarrow G \). Digitized images with \( l = 256 \) gray levels are considered in this thesis.

Let

\[ D_k = \{ (x, y) : I(x, y) = k, (x, y) \in D \}, \]

\( k = 0, 1, ..., l - 1 \), and

\[ h_k = \frac{n_k}{N \cdot M}. \quad (4.1) \]
\(k = 0, 1, ..., l - 1, \quad k = 0, 1, ..., l - 1, \) where \(n_k\) denotes the number of pixels in \(D_k\).

Obviously, \(\bigcup_{k=0}^{l-1} D_k = D\) and \(D_j \cap D_k = \phi\) for \(j \neq k\), where \(\phi\) represents an empty set. Thus, \(0 \leq h_k \leq 1, \quad k = 0, 1, ..., l - 1, \quad \sum_{k=0}^{l-1} h_k = 1\) and \(H = \{h_0, h_1, ..., h_{l-1}\}\) is the histogram of the image. Therefore, \(\Pi_l = \{D_0, D_1, ..., D_{l-1}\}\) is a PP of \(D\) with a probabilistic distribution

\[
p_k \equiv P(D_k) = h_k, \tag{4.2}
\]

where \(k = 0, 1, ..., l - 1\). This means that an \(l\)-level digitized image is characterized by the PP and \(\Pi_l\) of its domain derived from Equations 4.1 and 4.2.

### 4.2 Probability Partition

For an original image with 256 gray levels, a pixel \((x, y)\) with gray level \(f(x, y) = 0\) is “black” and is assumed to belong to the “object” class, while a pixel \((x, y)\) with gray level \(f(x, y) = 255\) is “white” and is assumed to belong to the “background” class. For pixels \((x, y)\) with other gray levels, i.e. \(0 < f(x, y) < 255\), are assumed to be in the gray area between these two classes. In other words, we can say they partly belong to the “object” and partly belong to the “background”. In this Chapter, we deal with how to select a threshold for bi-level thresholding. Let \(t\) \((0 < t < 255)\) be a threshold. The two partitions of \(D\) generated by \(t\) is given as:

\[D_d = \{(x, y) : f(x, y) \leq t, (x, y) \in D\}\]
\[ D_b = \{(x, y) : f(x, y) > t, (x, y) \in D\} \]

with probability distributions

\[ p_d(t) = P(D_d) = \sum_{k=0}^{t} h_k; \]

and

\[ p_b(t) = P(D_b) = \sum_{k=t}^{255} h_k. \]

Thus, the probabilities of the pixels of \( D \) belonging to classes “object” and “background” are \( p_d(t) \) and \( p_b(t) \), respectively.

### 4.2.1 Fuzzy Probability

Fuzzy probability of class \( A \) is defined as:

\[ p_A = \mu_A \cdot P(A) \quad (4.3) \]

where \( \mu_A \) is the fuzzy membership of class \( A \), and \( P(A) \) is the probability of class \( A \).

\[ p_d(t) = P(D_d) = \sum_{k=0}^{t} h_k; \]
Because of the relationship between Fuzzy Partition and Probability Partition, the task of looking for the optimal fuzzy sets which generate the maximum entropy becomes one to look for the fuzzy set which gives the corresponding fuzzy probability partition.

4.3 Thresholding Method based on Fuzzy Partition and Entropy

Cheng, Chen and Li [55] propose a method based on fuzzy partition and maximum entropy principle. An image is viewed as a fuzzy event modeled by a probability space. For two level thresholding, the pixels with gray level \( l \) can belong partially to the background class and the object. The probability of the pixels with gray level \( l \) belonging to a certain class is represented by its fuzzy membership function \( \mu \) as shown in Figure 4.1. The membership function can have different shapes, but for simplicity it is usually assumed to be a function as shown in Figure 4.1.

The fuzzy membership functions of the two classes, background \( \mu_b(k) \) and object \( \mu_d(k) \), in two-level thresholding are defined as below:

\[
\mu_d(k) = \begin{cases} 
1 & k \leq a \\
\frac{c-k}{c-a} & a < k \leq c \\
0 & k > c
\end{cases}, \quad (4.4)
\]

\[
\mu_b(k) = \begin{cases} 
0 & k \leq a \\
\frac{k-a}{c-a} & a < k \leq c \\
1 & k > c
\end{cases}. \quad (4.5)
\]
The threshold value $T$ is the crossover point of the fuzzy sets.

$$T = (a + c)/2.$$ 

How to search the optimal fuzzy sets that are associated with the maximum entropy efficiently is the next question. The exhaust searching method is used in this method for two level thresholding and the simulated annealing algorithm is used for multi-level thresholding.

**Exhaust Searching**

Exhaust searching, as its name suggests, finds the optimal parameter by trying every possible combination out of the whole searching database. In an 8-bit gray scale image, there are 256 gray levels. For $n$-level thresholding, we need $n - 1$ pair of membership functions, which have
\[
\binom{256}{2(n-1)} = \frac{256!}{(2(n-1))!(256 - 2(n-1))!}
\]

possible combinations. We can find the optimal membership functions by comparing the entropy of each thresholded image using the thresholds generated by each set of membership functions. But when \( n \) increases, even for \( n = 3 \), the possible combinations are approximately \( 1.7 \times 10^8 \). It is unrealistic to compute the entropy generated by every possible fuzzy partition. For \( n = 2 \), we can use the exhausting search to find the optimal fuzzy sets. But this is obviously not efficient.

4.4 Fast Searching Scheme

4.4.1 Description of the search procedure

Our goal is to find the best possible set of \((a, c)\) as fast as possible (see Figure 4.2).

Let us look at Figure 4.2, where \( h(x) \) is a histogram function in terms of the gray scale \( x \), and \( \mu(x) \) is the membership function in terms of the gray scale \( x \). The \( y \) axis represents different meanings for \( h(x) \) and \( \mu(x) \). For the former it represents the percentage of the number of pixels with a certain level in one image in terms of all pixels. The sum of \( h(x) \) equals 1, which is \( \sum_{x=0}^{255} h(x) = 1 \). While for the latter, it represents the fuzzy membership of the pixels with a certain gray level. Their value is from 0 to 1. But the sum of \( \mu(x) \) does not equal to 1. We put these two functions together into one diagram in order to illustrate the searching procedure more clearly.
We know that there are \( n \ (n = \binom{255}{a}) \) sets of possible combination of \((a, c)\). It is our goal to find the one which will derive the best entropy efficiently. So let’s consider any set with parameter \((a, c)\). For any \(a\), there are \(m\) \((m = 255 - a + 1)\) possible values of \(c\) \((a \leq c \leq 255)\). The membership function \(\mu(x)\) is shown as in Figure 4.2, as well as the histogram \(h(x)\). According to the definition of fuzzy probability (see Equation 4.3), the fuzzy probability \(P\) generated by this set of \((a, c)\) equals the sum of area A (because \(\mu(x) = 1\) in this period), and production of area B and the membership function \(\mu(x)\). Mathematically, it can be written as

\[
P = A + B \ast \mu(x).
\]  

(4.6)

If \(a\) is fixed, for one image (which means the histogram is fixed) area A is fixed. For \(a \leq x \leq c\), \(\mu(x)\) increases with the increase in the value of \(c\). Thus, so does the value
of $P$ increase. Area $A$ can be calculated according to the equation below:

$$A = \sum_{x=0}^{a} h(x),$$

because $\mu(x) = 1$, for $0 \leq x \leq a$.

$$B = \sum_{x=a+1}^{c} h(x),$$

Equation 4.6 can be written as:

$$P = f(a, c) = \sum_{x=0}^{a} h(x) + \sum_{x=a+1}^{c} (h(x) \cdot \mu(x)).$$  \hspace{1cm} (4.7)

To start searching, the value of $c$ is usually randomly chosen between $a$ and 255. If the calculated result is not equal to the required value, we need to move the value of $c$ backwards or forwards, depending on whether the calculated result is greater or lower than the one required. If greater, $c$ is moved backwards to $c_1$, otherwise, $c$ is moved to $c_2$.

For a two-level thresholding, we only need to search one fuzzy set $(a, c)$ which derives the fuzzy probability $p = \frac{1}{2}$. Theoretically, if a histogram function $h(k)$ is continuous in the range $\{k : 0 \leq k \leq 255\}$, and $a, c$ are real numbers, instead of integers, there exist innumerable sets of $(a, c)$ which satisfy the condition $P = f(a, c) = \frac{1}{2}$. However, because a histogram function is discrete and $(a, c)$ can only be a pair of integers, there are a limited number of sets $(a, c)$ if any, which perfectly satisfy the condition $P = \frac{1}{2}$. So the goal of our search is to find the possible best set which gives the maximum entropy. The search procedure is similar to Newton's method of approximation of root finding. Suppose the fuzzy probability $P$ is
a function of fuzzy set $S(a, c)$, which can be written as $P = f(S)$. In this case, the problem becomes finding the root of equation: $f(S) - \frac{1}{2} = 0$.

4.4.2 Search Procedure

According to the discussion above, the searching procedure can be described as below:

1. Input Image.

2. Calculate the histogram $h(k)$ of the image.

3. Initialization: Set $a = 0, c = 255; p = \frac{1}{2}$. Output state $a_{output} = c_{output} = 0; p_{output} = 0$.

4. For ($a = 0; a \leq 255; a + +$) : Compute $F(a) = \sum_{k=0}^{a} h_k$. Check:
   
   i. If $F(a) < p$, start search loop LOOP1;

   ii. Otherwise, check if $|F(a) - p| < |p_{output} - p|$, update output state:

   
   

   

   a_{output} = c_{output} = a, p_{output} = F(a), and exit; otherwise, exit the search procedure.

   * LOOP1 Set $c_0 = 255, c_- = a, c_+ = 255$. $c_-$ and $c_+$ are the lower and upper limit of the search range for $c$, respectively. Then,
(A) Calculate fuzzy probability \( p_d \) by Equation 4.3.

\[
p_d = \sum_{x=0}^{255} h(x) \cdot \mu(x), \text{ where } h(x) \text{ is the histogram and } \mu(x) \text{ is calculated according to Equation 4.4.}
\]

(B) If \( p_d < p \), set \( c_- = c_0 \); if \( p_d > p \), set \( c_+ = c_0 \); otherwise, output state \( a_{\text{output}} = a, c_{\text{output}} = c, p_{\text{output}} = p \), exit.

(C) Check search space \( s = c_+ - c_- \). If \( s < 2 \), check the current state and update output state if necessary; otherwise, set \( c_0 = (c_+ + c_-)/2 \), go to LOOP1.(A).

The flow chart of the search procedure described above is shown in Figure 4.3.

4.5 Experiment Results

4.5.1 Fuzzy Searching VS. Exhaust Searching

The proposed method is tested on many images to verify its accuracy and efficiency. To test the accuracy of this fast searching method, we compare the optimal fuzzy set obtained through our searching method with the one obtained through the exhausted searching method. We know that the exhausted searching method goes through every set in the searching space (all possible combinations of \( a \) and \( c \)). The fuzzy set obtained from the exhausted searching is the best possible set which derives the maximum entropy. We check if the fast searching method gives us the same set.
Figure 4.3: Flow chart of the search procedure for fuzzy-2 partition
evaluate its efficiency, we compare the computation time consumed by each method. Although the computation time depends on the type of computer as well as programming skill, it can still give us a rough idea of the difference between the searching speed of each method. Also the probability partition generated by the fuzzy set and the derived entropy are examined to see how good these fuzzy sets are.

Figure 4.4 to Figure 4.6 are some examples of the images on which experiment is carried out. Figure 4.4 (a) shows a grayscale image of a submarine in the sea. The 2-level thresholded image is shown in Figure 4.4 (b), and Figure 4.4 (c) shows the histogram of the grayscale image and the fuzzy partition of the histogram. The optimal fuzzy partition searched through our searching scheme is 
\[(a, c) = (86, 160),\]
and the threshold is \[t = 123.\] Same as Figure 4.4, Figure 4.5 and Figure 4.6 (a), (b) and (c) show the original grayscale images, 2-level thresholded images and histograms with the obtained Fuzzy Partition.

The experiment is carried out on a PC (Pentium III 1G). searching times for each of these three images are similar using exhaust searching scheme because the searching space is the same (in the range of 0 to 255). The searching times for the three images are \[181ms, \ 180ms \text { and } \ 180 \text { ms}\] respectively; while using the fast searching scheme the searching times are \[10ms, \ 20ms \text { and } \ 20ms\], respectively. Both methods obtain the same fuzzy set, indicating the same threshold. Thus, the fast searching scheme is demonstrated to be both accurate and efficient. This efficiency
makes it possible to extend the fuzzy entropy method to multiple level thresholding which are explored further in the later chapters.

4.5.2 Fuzzy Entropy VS. Entropy

As discussed in Chapter 3 Section 3.3.2, fuzzy entropy is derived from fuzzy probability. Compared to the original definition of entropy [27], fuzzy entropy shows more flexibility in image thresholding than it does with its original definition. Figures 4.7 to 4.9 demonstrate the thresholded images of Figures 4.4(a) to 4.6(a) by using Pun’s two methods in which the original definition of entropy is applied.

4.5.3 Fuzzy Entropy VS. Other Entropy Methods

Figures 4.10 to 4.12 show the 2-level thresholded images after applying (a): Kapur’s method, (b): Pal and Pal’s first method, (c): Pal and Pal’s second method and (d): Johannsen and Bill’s method. The corresponding thresholds derived from each method and their relationship with the histogram of the images are also shown in these figures (right). We notice that the part corresponding to Johannsen and Bill’s method is not shown in Figure 4.12 because their method fails to produce a sensible threshold for the test image.
Figure 4.4: Fuzzy 2-partition with a submarine image. (a): Original grayscale image; (b) 2-level thresholded image; (c) Histogram of the original image and the fuzzy set: a=86, c=160. Threshold: t=123.
Figure 4.5: Fuzzy 2-partition with a building image. (a) Original grayscale image; (b) 2-level thresholded image; (c) Histogram of the image and the fuzzy partition: $a=118$, $c=167$, threshold $t=142$. 
Figure 4.6: Fuzzy 2-partition with a kiosk image (a) Original grayscale image; (b) 2-level thresholded image; (c) Histogram of the image and the fuzzy partition: a=34, c=252, threshold t=143.
Figure 4.7: 2-level thresholded image of Figure 4.4(a) by using (a): Pun’s first method. *Threshold* = 119; (b): Pun’s second method. *Threshold* = 120.
Figure 4.8: 2-level thresholded image of Figure 4.5(a) by using (a): Pun’s first method. *Threshold* = 145; (b): Pun’s second method. *Threshold* = 140.
Figure 4.9: 2-level thresholded image of Figure 4.6(a) by using (a): Pun’s first method. Threshold = 159; (b): Pun’s second method. Threshold = 167.
Figure 4.10: (Left): 2-level thresholded image of the submarine by (a) Kapur’s method; (b) Pal and Pal’s first method; (c) Pal and Pal’s second method; (d) Johansen and Bill’s method. (right): Histogram and the corresponding threshold obtained from the four methods.
Figure 4.11: (Left): 2-level thresholded image of the building by (a) Kapur’s method; (b) Pal and Pal’s first method; (c) Pal and Pal’s second method; (d) Johansen and Bill’s method. (right): Histogram and the corresponding threshold obtained from the four methods.
Figure 4.12: (Left): 2-level thresholded images by (a) Kapur’s method; (b) Pal and Pal’s first method; (c) Pal and Pal’s second method. (right): Histogram and the corresponding threshold obtained from these methods.
4.5.4 S Function VS. Linear Function

Another commonly used function which represents the membership of a fuzzy set is the S-function which is defined as below:

\[
S(x; a, b, c) = \begin{cases} 
0, & x \leq a \\
\frac{(x-a)^2}{(b-a)(c-a)}, & a < x \leq b \\
1 - \frac{(x-c)^2}{(c-b)(c-a)}, & b < x \leq c \\
1, & x > c
\end{cases}
\] (4.8)

where \(a, b\) and \(c\) are parameters which determine the shape of the S-function. Figure 4.13 shows an S-function used in the fuzzy partition. Our experiment with both functions shows that they produce almost the same thresholds.

![Figure 4.13: S function](image)
4.6 Summary

In this Chapter, a 2-level thresholding technique for a digitized image is proposed. This method is derived using the maximum entropy principle and the fuzzy partition. The entropy function is defined by the probabilistic partition (PP) and the fuzzy 2-partition (FP) of the domain of a given image. The FP is built by applying Bayes’ formula to each part of the PP. The value of the entropy function is used to measure the compatibility between the PP and the FP. For a given image, the PP is directly obtained from the histogram of the image, while the FP is unknown. The required FP is found when the entropy function arrives at the maximum value, and the best threshold value is finally obtained from the required FP. A necessary condition for the entropy function arriving at the maximum value is derived. Next, a technique is proposed based on the condition that a search be carried out of the required FP. Experiment results show that the proposed method is just as powerful as the exhaust searching method for accuracy, but much faster in the searching speed. This facilitates the extension of the fuzzy thresholding method to multilevel thresholding.