Chapter 8
Adaptive Fuzzy Thresholding for Color Images

Color images usually contain more information than gray scale images. Uniform areas can be extracted better by applying the thresholding method to color images than to gray scale ones. Figure 8.1 shows different segmentation results by applying thresholding methods to gray scale and color images, respectively. Figure 8.1(a) is a gray scale image and its original color image is shown in 8.1(d). We can see that, although there are clear differences between the colors of the baby skin, the pot and the background, the contrast among the intensities of these colors is not that distinct in the gray scale image. After applying the fuzzy-2 partition method in Chapter 4 we obtained its 2-level thresholded image (shown in 8.1(b)). We can see that the baby’s face and body have been distinguished from the shadow of the hat but merged with the background and the front of the pot. We tried to expand the threshold level to see if the different areas can be separated. The three- and five-level thresholded image using the methods in Chapter 5 and Chapter 6 on the gray scale image are shown in 8.1(c) and (f). In the three-level thresholded image the pot and the baby are distinguished but the background is still mixed with the baby. In the five-level image, although these three are assigned with different colors, the over-segmentation in the gray scale space causes many fractures in the baby’s face area.
Figure 8.1: The color image contains more information than the gray scale image and the unique area can be distinguished better by using the thresholding method on the color image. (a) a gray scale image; (b) its 2-level thresholded image using fuzzy-2 partition method; (c) its three-level thresholded image using fuzzy-3 partition method; (d) the original color image of image (a); (e) the thresholded image using fuzzy-2 partition on the color image; (f) 5-level thresholded image using fuzzy-5 partition method on the gray scale image.
This oversegmentation in one color dimension can be avoided by using partition in the three-dimension color space. 8.1(e) shows the thresholded image after using the fuzzy-2 partition method with the color image. The three areas are clearly extracted and the face features are well preserved.

The application of the fuzzy c-partition can be extended to color image thresholding. The basic concept is to simply apply the thresholding in each of the R, G and B dimensions. Then combine them by crossing the segmentations in each dimension. First let us look at the situations with two and three level thresholdings. Figure 8.2 shows the images used in the experiment.

![Color images used for the test](image)

Figure 8.2: Color images used for the test

### 8.1 Simple Extension of Fuzzy Partition

#### 8.1.1 Two($2^3$)-Level Thresholding

We know that color images can be decomposed into three images—the images in R (red), G (green) and B (blue) dimensions, respectively. Our first step in applying
fuzzy partition to color images is to segment the three decomposed RGB images with the method separately and then combine the segments by crossing them. For example, if the histogram of the image on the R dimension is divided into two parts, represented by “0\textsubscript{R}” and “1\textsubscript{R}”, so are the histograms of G and B. We get “0\textsubscript{G}”, “1\textsubscript{G}”, “0\textsubscript{B}” and “1\textsubscript{B}”. Put these segmentations into 3D, we get \(2^3 = 8\) groups. They are \((0\textsubscript{R}, 0\textsubscript{G}, 0\textsubscript{B})\), \((0\textsubscript{R}, 0\textsubscript{G}, 1\textsubscript{B})\), \((0\textsubscript{R}, 1\textsubscript{G}, 0\textsubscript{B})\), \((0\textsubscript{R}, 1\textsubscript{G}, 1\textsubscript{B})\), \((1\textsubscript{R}, 0\textsubscript{G}, 0\textsubscript{B})\), \((1\textsubscript{R}, 0\textsubscript{G}, 1\textsubscript{B})\), \((1\textsubscript{R}, 1\textsubscript{G}, 0\textsubscript{B})\) and \((1\textsubscript{R}, 1\textsubscript{G}, 1\textsubscript{B})\), respectively. Thus, by dividing histograms of each dimension into two parts we obtain eight levels of segmentation of the color image.

For example, Figure 8.3(a) shows a color image. Figure 8.3(c)-(e) shows its histograms in R, G and B dimensions and the Fuzzy-2 Partitions on these histograms. In each dimension we get two classes. Combine them and we get eight classes. The thresholded image is shown in Figure 8.3(b).

This method is tested on the color images shown in Figure 8.2 and the experiment results are shown in Figure 8.4. The means of the RGB value of pixels in each group are calculated to represent the color of each group to make the segmented image closer to the original one.
Figure 8.3: (a) Color image; (b) Thresholded image (8 colors); (c) Fuzzy-2 Partition in R dimension. \((a, c)^R = (17, 143)\) and \(T_R = 80\); (d) Fuzzy-2 Partition in G dimension; \((a, c)^G = (45, 160)\) and \(T_G = 102\); (e) Fuzzy-2 Partition in B dimension; \((a, c)^B = (95, 104)\) and \(T_B = 99\).
8.1.2 Three\(^{3}\)-Level Thresholding

Similar to two-level thresholding, three-level thresholding for color images is to segment each of the three histograms (histograms on R, G, B dimension) into three parts and then cross them to get \(3^{3} = 27\) levels of the color image.

Figure 8.5 shows the thresholded color images with fuzzy 3-partition.

Figure 8.5: Thresholded color images with Fuzzy-3 partition.
8.2 Adaptive Color Image Thresholding with the Quadtree Scheme

The experiment results of the simple extension of Fuzzy-2 Partition and Fuzzy-3 Partition (See Figures 8.4 and 8.5) are not very impressive because too many color classes (8 levels in Fuzzy-2 and 27 levels in Fuzzy-3) may cause over-segmentation in the color space. Some classes may only contain a few pixels. Thus, it may help to have the validity of each class checked after the segmentation. And also, applying the quadtree scheme will help to avoid the segmentation of the background into different classes (As in Figures 8.4(d) and 8.5(d)).

The quadtree scheme for color image segmentation is similar to the one for gray scale images.

8.2.1 The Algorithm of Adaptive Thresholding

The algorithm is described below:

1. Input image and decompose it into three images in R, G and B dimensions.

2. For each image apply the quadtree scheme from Section 7.1. The quadtree structure is generated to separate the heterogenous areas from the homogeneous areas.

3. Calculate the histogram of the heterogeneous areas for each image $h^R_k$, $h^G_k$ and $h^B_k$. 
4. Apply fuzzy 2-partition on each histogram and search for the optimal fuzzy sets, 
\((a, c)^R, (a, c)^G\) and \((a, c)^B\).

5. Compute the threshold \(T\) for each image. 
\[ T^R = (\frac{a+c}{2})^R, \quad T^G = (\frac{a+c}{2})^G \] 
and 
\[ T^B = (\frac{a+c}{2})^B. \]

6. Segment each image into two classes with its corresponding threshold. By 
crossing them we get eight classes. For a pixel \(f(x, y)\) with RGB value \((r, g, b)\), 
the value in RGB space after thresholding is
\[
 f'(x, y) = \begin{cases} 
 (0, 0, 0) & \text{if } (r \leq T^R) \land (g \leq T^G) \land (b \leq T^B) \\
 (0, 0, 1) & \text{if } (r \leq T^R) \land (g \leq T^G) \land (b > T^B) \\
 (0, 1, 0) & \text{if } (r \leq T^R) \land (g > T^G) \land (b \leq T^B) \\
 (0, 1, 1) & \text{if } (r \leq T^R) \land (g > T^G) \land (b > T^B) \\
 (1, 0, 0) & \text{if } (r > T^R) \land (g \leq T^G) \land (b \leq T^B) \\
 (1, 0, 1) & \text{if } (r > T^R) \land (g \leq T^G) \land (b > T^B) \\
 (1, 1, 0) & \text{if } (r > T^R) \land (g > T^G) \land (b \leq T^B) \\
 (1, 1, 1) & \text{if } (r > T^R) \land (g > T^G) \land (b > T^B) 
\end{cases}
\]

7. Check the validity of each class.

(a) Calculate the number of pixels in each class.

(b) If the number of pixels in a class is below a certain value \(N^\ast\), then this class 
is not a valid class. The pixels in this class are classified into the closest 
class according to their distance in the RGB space.

8.2.2 Experiment Results

Some experiment results are shown in Figure 8.6. The number of colors in each 
thresholded image is 4, 2, 3 and 3, respectively. We can see that the number of
colors used in the segmented images vary depending on the color information in the different images. Colors are classified into several major groups according to the fuzzy 2-partition of each color dimension. The color groups containing a minority pixels of the whole image are considered to be an invalid class and merged into its closest large class. Fuzzy 2-partition is used instead of fuzzy three partition because eight-level thresholding is usually enough in many applications.

Figure 8.6: Experiment result with the adaptive color image thresholding method.

8.3 Summary

In this chapter, an adaptive fuzzy thresholding method for color images is presented. This method employs the quadtree scheme to separate the image into homogeneous and heterogeneous areas. The color histograms are computed for the heterogeneous area and used to calculate the optimal fuzzy sets. Fuzzy 2-partition is used on each of the three histograms in R(red), G(green) and B(blue) dimensions and by crossing the results in a 3D, eight-level segmentation is obtained. The validity of each of the eight
classes is checked and only the classes with a certain number of pixels are kept as a valid class. The rest of the classes are merged into their neighboring valid classes. Experiment results show that this method can determine automatically the number of the thresholds levels and achieves good results for images having large homogeneous areas.