EE-634 Digital Image Processing Term Project Report

Alican Hasarpa, Bora Baydar, Volkan Okbay

1-Introduction

Image segmentation is partitioning of a digital image into segments which are suitable for image analysis. The resulting image after image segmentation has more meaningful representation which provides better understanding of the image or easy to analyze. Image processing can be done by using different methods such as clustering based methods, edge based methods, compression based methods, region based methods, graph based methods, pixon based methods. It is considered as an important application of computer vision since it is widely used in areas such as medical imaging (locating tumors and other pathologies, virtual surgery simulation etc.), machine vision, video surveillance, pedestrian detection[4]. In this report, we will consider 3 image segmentation methods which are edge based segmentation, color based segmentation using k-means clustering and watershed transformation.

2.a-Edge Based Segmentation

Edge based image segmentation methods are based on finding the boundaries within the image. The boundaries form the edges which creates discontinuities in the resulting image [5]. It is important to note that the edges are generally formed between different objects, so this method is more suitable for object detection tasks. Finding the edges can be done by using different algorithms. They can be grouped into 2 main groups[6].

A) First order derivative edge detectors:

They are also called as gradient operators. These operators take the first order derivatives of the pixel values in the specific directions and merge the results in the edge map. Roberts, Prewitt and Sobel filters are most widely used edge detectors in this group.

B) Second order derivative edge detectors:

Unlike gradient operators, second order derivative edge detectors are used for locating the local maxima in gradient values and mark them as edges in the edge map. Laplacian of Gaussian filter and Canny filters are most famous and effective examples in this group. Although they are just a little bit more complex computationally than gradient ones, second order derivative edge detectors give better results for edge detection. It can be observed from the Figure 2.a.2, for original example image in Figure 2.a.1.

Figure 2.a.1

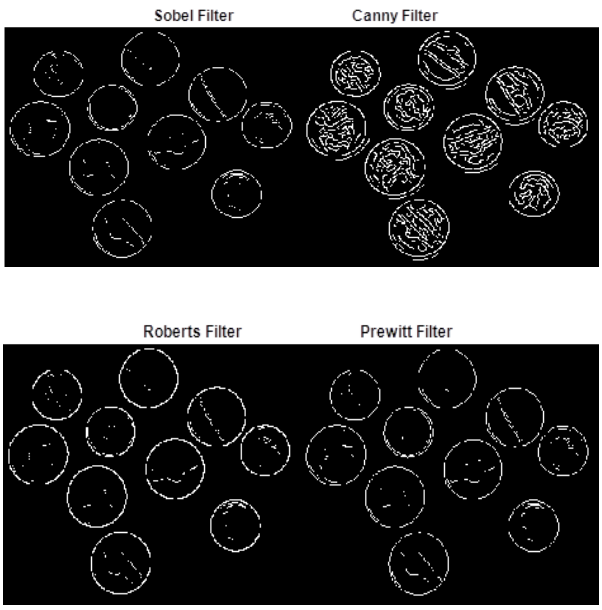

Figure 2.a.2

Because of its performance, we used Canny algorithm in our image segmentation implementation. Instead of using only R,G and B components of the original images, we also implemented the edge detection on the luminance component of the images which also has lots of details about the image. Finally, we merged them all in one binary image.

The biggest problem of using the edge detection for image segmentation is existence of the edges which are not continuous. In order to lengthen those edges and to form the continuous boundaries, we applied dilation by 4-pixel-disk and erosion after it by 3-pixel-disk. This operation extends the edges and emphasizes them. On the other hand, it will result with losing the important details when applied on the image with small objects. In that case, only the closing operation should be used on the image. After having the modified edge map, we decided to use an algorithm from region based methods. In the binary edge image, we implemented a labeling which satisfies 4-pixel-connectivity.

<u>4-connected-neighborhood of pixel-p[7]:</u>

1
4 p 2
3

Algorithm checks p; if it is 1 and any of its neighbors 1, 2, 3, 4 is also 1, then p is connected to these neighbors. For example, the algorithm results with 2 different objects for Figure 2.a.3; one is 2x2 and the other one is 2x3 size.

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

Figure 2.a.3

2.b-Color Based Segmentation Using K-means Clustering

K-means clustering is basically a way to partition a dataset into k clusters where k is a predefined number. K-means clustering has been widely used in image segmentation since 1990s [2]. The algorithm steps are given below:

1. First place k center points randomly
2. For each data point x, find the nearest center point
3. After each point is assigned to a cluster, find the new centroid of that cluster by averaging its member data points
4. Repeat 2nd and 3rd steps until centroids stop changing.

The Loyd's algorithm, which is mostly known as k-means algorithm, solves the following equation:

$$\arg\min_{\mathbf{c}} \sum_{i=1}^{k} \sum_{\mathbf{x}\in c_i} d(\mathbf{x}, \mu_i)^2 = \arg\min_{\mathbf{c}} \sum_{i=1}^{k} \sum_{\mathbf{x}\in c_i} \|\mathbf{x} - \mu_i\|_2^2$$

where k is the number of clusters, c_i is the group of points of i^{th} cluster. A simple example can be seen from Figure 2.b.1 :

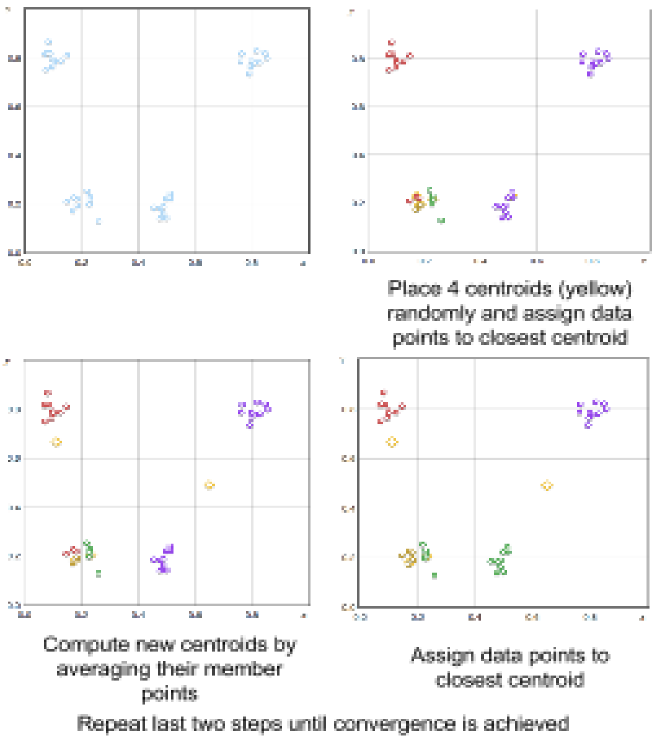

Figure 2.b.1

Its main disadvantage is that k (number of clusters) should be known priorly. In our proposed method, k is found automatically by finding the number of the peaks in the image histogram. For example, a histogram as shown in Figure 2.b.1 has 5 peaks if minimum peak width is taken greater than 10. Therefore, the image corresponding to that histogram would have 5 clusters.

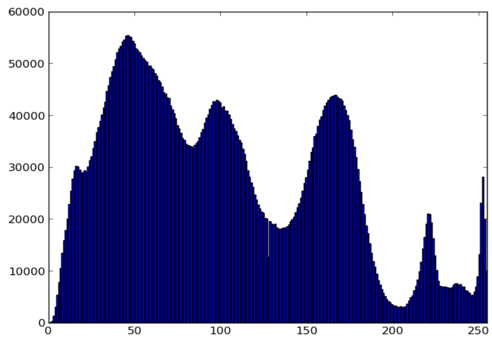

Figure 2.b.2

In our algorithm, we have used both L*A*B and HSV color spaces and observed their results. The included MATLAB code for k-means clustering runs with L*A*B space. These two color spaces are selected because they represent colors separated from lightness or saturation and also our literature survey showed us that these two color spaces work best with k-means clustering as pointed out in [1]. K-means clustering in Image Segmentation can also be thought as a histogram based segmentation since k-means algorithm is applied on 2D or 3D histograms of the image.

Our overall algorithm converts the original image to L*A*B or HSV colorspace, apply Gaussian Filter to smooth the image, take its histogram and find *k* by finding the number of clusters and apply k-means algorithm.

Although k-means clustering is a successful algorithm for image segmentation, it is computationally complex especially when k is great and 3 channels are used. In [3] the big O notation of k-means clustering is given as:

O(iterations*clusters*dimensions*instances)

2.c-Segmentation Using Watershed Transformation

Watershed concept comes from geodesic maps. They are defined by a catchment basins to which water drains and ridge line that is perimeter of the watershed. In fact, ridge lines are natural segmentations of watersheds.

In image processing we will use this concept as defining our object as catchment basins and try to acquire ridge lines that separates our objects or object features. In our application, inclination and peak-valley structure of geography corresponds to gray-scale intensity values. What matter is “local minima” in this case, which stands for objects. In the figure 2.c.1 below, there represented watershed ridge lines and catchment basins in a 3-D demonstration of an image.

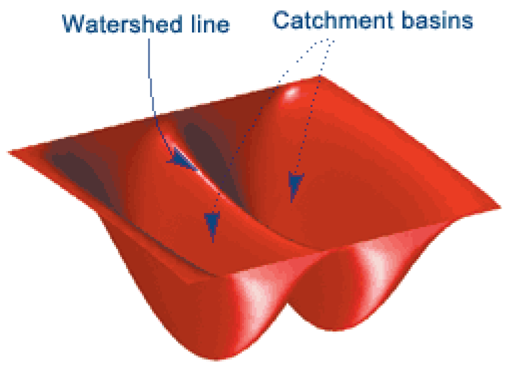

Figure 2.c.1

<u>Meyer’s Flooding Algorithm:</u> This mathematical approach originates on the inter-pixel watershed concept, coined by Beucher and Meyer (1991). Implemented as follows in general [8]:

1. Label local minima.
2. Non-labeled neighboring pixels to a priority queue.
3. Take out the pixel with least priority from the queue and check its neighborhood.
4. If all labeled pixels have the same label, give that label to current pixel. If there are more than one label then it is ridge line pixel.
5. Return step 2 if queue is not empty.

Here below we exemplified algorithm on a simple image matrix. Numbers stands for gray scale values (in figures 2.c.2-5)

3	4	4	4	1	3	4	4	4	1
3	2	5	4	4	3	2	5	4	4
3	3	3	3	3	3	3	3	3	3
3	2	2	0	1	3	2	2	0	1
3	1	1	1	2	3	1	1	1	2

3	4	4	4	1	3	4	4	4	1
3	2	5	4	4	3	2	5	4	4
3	3	3	3	3	3	3	3	3	3
3	2	2	0	1	3	2	2	0	1
3	1	1	1	2	3	1	1	1	2

Figure 2.c: (**2**) Label each minimum, (**3**) Choose low priority pixel from queue, (**4**) Check neighborhood and label, (**5**) Whole segmented image.

An average image has hundreds of local minima, so if watershed is applied without any processing, this will resulted in a more than desired number of segments, called "over-segmentation". To prevent that, before and after watershed application processing approaches would be needed in almost every case. That may be due to noise or camera blur; on the other hand, it can be needed to achieve a special purpose. Here below there are some popular methods and processes for a better watershed segmentation [9]:

- Noise removal
- Morphological operations (gradient, contrast, area filtering etc.)
- Low-pass or Gaussian filtering
- High-pass filtering, Edge-detection
- Local operations (mean, variance etc.)
- Distance Transform, Fourier Transform etc.
- Region Merging (Post-processing)

Those methods in fact, are fairly image and purpose-dependant. Some of them may cause to a slower algorithm implementation. By the way, watershed itself is a fast algorithm compared to other approaches. Besides region-merging, one or more pre-processing can be applied as stated above. Region-merging is a post-processing tool, which has variety of distinct ways to realize.

The most popular processing ways would be morphological operations. They are originated from dilation and erosion. Next level, opening and closing are widely used in background illumination techniques, object area filtering and so on [10]. In our project, more special functions of those are used: top- and bottom-hat filtering. Those two stands for below simple expressions and very handy in emphasizing peak-valleys, contrast adjustments, morphological gradient calculation and smoothing areas of object features (figure 2.c.6).

Figure 2.c.6 and 2.c.7

We have used a special and common approach to eliminate over-segmentation, namely "Marker Control". This method has a strong control over objects, which are desired to be segmented. In short, we will mark those points that we want to preserve local minima of. Besides our markers there will be no local minima remains. As a result, segments will be reduced to only object features we needed. In practice, gray-scale image is contrast adjusted and morphologically modified (figure 2.c.6-7). By thresholding we acquire marker image. Then impose those markers on the gradient image resulted by original gray-scale image (figure 2.c.8-9). In conclusion, we can apply watershed and get desired label matrix.

Figure 2.c.8 and 2.c.9

3. Conclusion

In consequence, watershed is a relatively fast, powerful in detecting desired features/objects, able to spot overlapped objects, can separate similar colored objects and combinable with every other method. But, as a significant downside, side processing can be expensive in time and performance. Method choice is also very image and purpose dependant. K-means clustering relatively slower but successful in detecting objects in a smoother way. Resultant images are more understandable by human viewer. This method is more color-dependant and threshold calculation algorithms vary. Edge-based algorithm, as a last approach, is the fastest among all. It detects objects easily but needs a careful selection of thresholds and graphical search. The application may not be that automated.

In conclusion, the choice affected by image properties and aim of the application. All the example results, timing, powerful and weak points of the algorithm should be considered. All resulting images and MATLAB(c) m files are separates files.

4. References

[1] Dibya Jyoti Bora, Anil Kumar Gupta, Fayaz Ahmad Khan, Comparing the Performance of L*A*B* and HSV Color Spaces with Respect to Color Image Segmentation, *International Journal of Emerging Technology and Advanced Engineering,* Volume 5, Issue 2, February 2015.

[2] Pappas T.N., Murray Hill, An adaptive clustering algorithm for image segmentation, *Signal Processing, IEEE Transactions*, Vol. 40, pp. 901-904, 2002

[3] http://www.cs.princeton.edu/courses/archive/spr08/cos435/Class_notes/clustering2_toPost.pdf

[4] Nikhil R. Pal, Sankar K. Pal**,** A review on image segmentation techniques, *Pattern Recognition,* vol. 26, pp. 1277-1294, 1993

[5] Ravi S and A M Khan, “Operators Used in Edge Detection: A Case Study”, International Journal of Applied Engineering Research, ISSN 0973-4562 vol. 7 No 11, 2012.

[6] N. Senthilkumaran and R. Rajesh, “Edge Detection Techniques for Image Segmentation – A Survey of Soft Computing Approaches”, International Journal of Recent Trends in Engineering, Vol. 1, No. 2, May 2009.

[7] http://blogs.mathworks.com/steve/2007/03/13/connected-component-labeling-part-2/

[8] Fernand Meyer. Un algorithme optimal pour la ligne de partage des eaux. Dans 8me congrès de reconnaissance des formes et intelligence artificielle, Vol. 2 (1991), pages 847–857, Lyon, France.

[9] Tanniru, Padmavati, "Effects of pre-processing and postprocessing on the watershed transform" (2007).Master's Theses.Paper 3390.

[10] http://utam.gg.utah.edu/tomo03/03_mid/HTML/node120.html

APPENDIX

Coding and resulting images given in separate files.