CENG466 FUNDAMENTALS OF IMAGE PROCESSING
THE3 REPORT

Emre Zinal
Computer Engineering
Middle East Technical University
2094837

Yunus Emre Gök
Computer Engineering
Middle East Technical University
2166460

***Abstract*—The purpose of this assignment is to get familiarize with the fundamental morphological image processing techniques and image segmentation.**

I. Object Counting

Image processing has a wide variety of applications from object detection to extracting depth maps from image and video streams. One of the important application areas of image processing is to count objects automatically from the given image or video. This application has a wide variety of use in industry. For example one can count the number of cells in a microscope image. With the advance of these algorithms computers are now able to separate and count objects with fast and highly accurate results.
In our homework assignment question 1, we are given different photos of flying jets. And we are expected to write the code to detect and count these jets automatically. Our code shows an image of the jets that it found with white on black background (as a binary image) and also writes the number jets found to the console.

A. Morphology

1) Dilation: $A \oplus B = \{z_i(B)_z \cap A = \emptyset\}$
2) Erosion: $A \ominus B = \{z_i(B)_z \subseteq A\}$
3) Opening: $A \circ B = (A \ominus B) \oplus B$
4) Closing: $A \bullet B = (A \oplus B) \ominus B$

B. Algorithm

1) Binarizing The Image: To apply the Minkowsy Operation we first converted the image from RGB to Binary with specified thresholds. In very image except 'A2.png', planes have darker colors in overall image. Hence, one threshold is enough to divide the image. However, In A2 , forest has darker colors. Therefore, we used two thresholds to separate the planes. Then we take complement of the images to apply operations.

2) Structuring Element: Minkowsky operations need a structuring element for probing and reshaping. We used spheres with different radiuses depends on the image and plane sizes.

3) Operations: Since objects in the images don't have uniform color distribution, some bays and holes occurred. Therefore, we used dilation to fill the gaps with radius 2. For filling large holes or bays, closing operation can be used.
After the dilation, We used erosion to separate the objects. We used spheres with different radiuses depends on the image because in some images planes are closer to each other and some of them has some noise or parts of other objects.
We used dilation again to connect the parts of the plane that are seperated by erosion. Structuring element is again sphere with different radiuses to connect the divided part since some of the planes has bigger size and the gaps are larger in these images.

4) Counting: Until now, we separated the planes from the rest of the image and divided them one. Final part is counting the connected parts.

C. Functions

1) imbinarize: MATLAB has its own binarize function with specified threshold.

$$BW = imbinarize(I, T)$$

Where I is the input image, T is threshold level ranged in [0,1] and BW is the result binary image.

2) threshold: We wrote a threshold function to binarize the image between two values.

$$res = threshold(I, t1, t2)$$

Where I is input image , t1 and t2 are the thresholds and res is the result image.

3) strel: MATLAB has strel function that produces structuring element to use in morphological operation.

$$SE = strel(type, parameters)$$

Where type is specifies shape like sphere or neighborhood. And parameters are for the sizing.(r for radius or w for width)

4) imdilate: MATLAB has its own function that dilates the grayscale or binary image.

$$J = imdilate(I, SE)$$

Where I is the input image and SE is the structuring element. J is the result image.

5) imdilate: MATLAB has its own function erodes the the grayscale or binary image.

$$J = imderode(I, SE)$$

Where I is the input image and SE is the structuring element. J is the result image.

6) bwconncomp: MATLAB has its own function to find connected component in a binary image.

$$CC = bwconncomp(BW)$$

$$CC = bwconncomp(BW, conn)$$

where BW is the binary image and CC is the connected components. conn is optional which specifies the connectivity.

7) counter: We wrote counter that counts the planes in the given image with specified parameters and writes the image of pixels groups that is counted.

$$counter(img, l1, l2, r1, r2, name)$$

where img is the input image, l1 and l2 is the threshold levels, r1 is the radius of structuring element of erotion and r1 is the radius of structuring element of dilation

D. Parameters

As we stated before we used different parameters for different images.

	Threshold 1	Threshold 2	Erosion Radius	Dilation Radius
A1	0.5	0	7	7
A2	0.4	0.22	10	30
A3	0.2	0	7	3
A4	0.2	0	7	3
A5	0.25	0	7	40
A6	0.008	0	4	22

1) Threshold 1: The difference on the thresh holds depends on the color distributions. Where A1 planes much more darker than the the background, in A6 there are very close colors to the planes.

2) Threshold 2: In very image except A2, planes have darker colors in overall image. Hence, one threshold is enough to divide the image. However, In A2 , forest has darker colors. Therefore, we used two thresholds to separate the planes. Others have only one threshold.

3) Erosion Radius: Erosion radius depends on the plane and noise sizes. In A2, taking the threshold did not separate the image perfectly and forest edge maintained its position. Therefore the erosion radius is higher to delete the edge.
Also, In A6 some planes has small sizes and larger erosion radius can make these disappear. Hence, radius is small. On the other hand , this situation caused the problem part of the mountain counted as a plane.

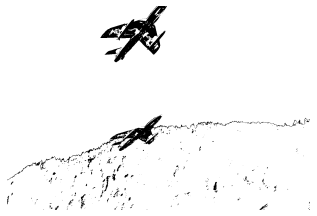

Fig. 1. Binarized A2

4) Dilation: Dilation radius depends on the distance between planes and the sizes of the planes. To join part of the planes in A5 radius should be larger.However In A4, making radius higher can reunite different planes and it can be counted as 1.

II. Segmentation

Image segmentation is the operation of partitioning an image into a collection of connected sets of pixels.[1] The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze.[2] By dividing the image into segments, we can process the important parts rather than whole image where some parts may contain no important information. In image segmentation we group together the pixels with similar attributes.
For this part we had to choose 2 different image segmentation algorithms that we learned in the lectures, and we had to apply them to the given pictures. We chose k-means and region-growing algorithms.

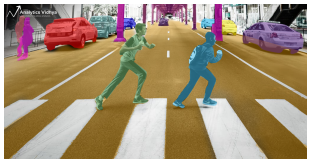

Fig. 2. Image segmentation example[3]

A. K-means Algorithm

K-means algorithm is a type of clustering algorithms. A cluster refers to a collection of data points aggregated together because of certain similarities.[4] The aim of clustering is to keep similar objects in the same cluster.
K-means algorithm works as follows:
1) First we need to define number of centroids we need.
2) Every point of the data set is assigned to its nearest centroid.
3) In other words, the centroid is moved toward the center of its assigned points.
4) Program halts creating and optimizing clusters when either:
-The centroids have stabilized — there is no change in their values because the clustering has been successful.
-The defined number of iterations has been achieved.[4]

Here in figure 2, we see the result of applying k-means

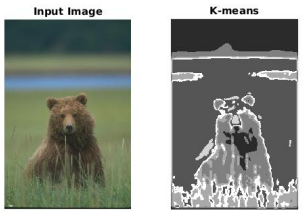

Fig. 3. K-means algorithm result

clustering algorithm for image segmentation on 100080.jpg that is given to us. We can clearly see that after applying segmentation, the shape of bear is extracted from the image. We selected number of centroids in the beginning as 6 (nClass variable in code) because there seemed to be at most 6 different regions at he images given.

In the k-means algorithm if the number of centroids which

Fig. 4. Effect of low number of centroids in k-means

is initially given as a parameter gets lower (to 2), we get the result in figure 4. We clearly see that the number of clusters found have dropped compared to what we found in figure 3. For some pictures this might result in some important regions to not found by the algorithm.

In compared to figure 4 if we increase the number of centroids to 12 in the code, we get the result in which more different regions found. This is not a good effect since with increasing number of centroids we find much more different clusters and some of them are actually on the same object but seems as different objects.

B. Region Growing Algorithm

Region growing is a simple region-based image segmentation algorithm. The fundamental drawback of histogram-based region detection is that histograms provide no spatial information (only the distribution of gray levels).Region-growing approaches exploit the important

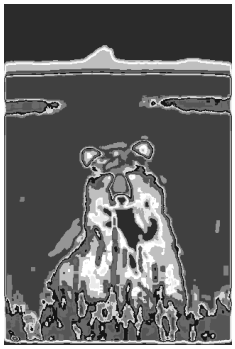

Fig. 5. Effect of high number of centroids in k-means

fact that pixels which are close together have similar gray values.[5] This approach to segmentation examines neighboring pixels of initial seed points and determines whether the pixel neighbors should be added to the region.[6] In region growing algorithm, we have the concept of seed points. We determine the seed points at first step. Seed point selection is based on some user criterion (for example, pixels in a certain grayscale range, pixels evenly spaced on a grid, etc.). The initial region begins as the exact location of these seeds. The regions are then grown from these seed points to adjacent points depending on a region membership criterion. The criterion could be, for example, pixel intensity, grayscale texture, or color. Since the regions are grown on the basis of the criterion, the image information itself is important. For example, if the criterion were a pixel intensity threshold value, knowledge of the histogram of the image would be of use, as one could use it to determine a suitable threshold value for the region membership criterion.[6]

After seed points and criteria selected,iteration continues based on selected criteria until there are no change in two successive iterative stages.

For the second part of our assignment, we applied region

Fig. 6. Region growing algorithm result

growing algorithm for all 25 of the images given. We selected seed points at middle of every image since we determined

the main objects are at the middle of every image generally. Here in figure 6, the region growing algorithm correctly defines the boundary of the bird and tree. It also distinguishes background from them.

In the code we submit, difference between a pixel's intensity value and the region's mean, is used as a measure of similarity. The pixel with the smallest difference measured this way is allocated to the respective region.

As indicated in the assignment, the code authors are defined at the beginning of each file.

REFERENCES

[1] http://www.cs.bilkent.edu.tr/ saksoy/courses/cs484-Fall2019/index.html
[2] https://en.wikipedia.org/wiki/Image_segmentation
[3] https://www.analyticsvidhya.com/blog/2019/04/introduction-image-segmentation-techniques-python/
[4] https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1
[5] https://www.cse.unr.edu/ bebis/CS791E/Notes/RegionGrowing.pdf
[6] https://en.wikipedia.org/wiki/Region_growing