ISSN:2231-0711
www.ijcset.net

Optimization of the Edges Detected by Canny Operator through Segmentation

Megha Seth, Dr. Sipi Dubey

Rungta College of Engg. & Tech.
Bhilai, Chhattishgarh,India
meghaseth1983@gmail.com
drsipidubey@gmail.com

Abstract— **A problem that occurs while applying the canny operator directly for edge detection on an image that consist of large number of boundaries is that their output becomes more complex because of large number of edges and due to complexity finding out a particular region of an image becomes very difficult. In this paper we are using K-means clustering algorithm to optimize the output of canny operator by segmenting the image into number of segments given by user as per the requirement for the application and then apply canny operator on the segmented image.**

Keywords— **Image segmentation, Global segmentation, Local segmentation Clustering, K – Means, Canny Operator, Centroid, RMSE.**

I. Introduction

Image segmentation [1][2] is a important research area in digital image processing and it is extremely large area, often depending on a variety of techniques from a wide range of other mathematical fields. Segmentation is the process of partitioning /subdividing a digital image into multiple meaningful regions or set of pixels regions with respect to a particular application that do not overlap. The segmentation is based on measurements taken from the image and might be grey level, colour, texture, depth or motion. The result of image segmentation is a set of segments that collectively cover the entire image. All the pixels in a region are similar with respect to some characteristics or computed property, such as colour, intensity or texture. Segmentation can be categorized as:

Global Segmentation: It is concerned with segmenting a whole image.
Local Segmentation: It deals with segmenting sub-images which are small windows of a whole image or we can say that it is a fragment of a larger scene being processed in isolation.
Region growing [4] and clustering are two representative methods of region-based segmentation. Drawbacks of the region growing method are that it is difficult to make the growing or stop growing criteria for different images and the method is sensitive to noise. Recently, most researchers focus on treating the segmentation problem as an unsupervised classification problem or clustering problem. In their methods, segmentation is obtained as the global minima of criterion functions associated with the fuzzy / possibilistic distance between the prototypes and the image pixels. By partitioning pixels according to their global feature distribution, these methods achieve good global partitioning results for most of the pixels. But in these methods, spatial relationship of the pixels is rarely considered. The loss of spatial information of the pixels maybe leads to unreasonable segmentation results for that the pixels that are similar in low level feature (color etc.) but separate in spatial will be grouped into one region. And at the same time, run time complexity of this global partition is often high.
The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. Clustering is an example of unsupervised learning.

II. K-Means algorithm

K-Means [6] is a simple but well known unsupervised learning algorithm for grouping objects. All objects needs to be represented as a set of numerical features. The algorithm has as an input a predefined number of clusters which is given in k. The 'means' stands for an average, an average location of all the members of a particular clusters. The K-means algorithm is an iterative procedure where final results depend on the values selected for centroid. A centroid is an artificial point in the space of records which represents an average location of the particular cluster.
The main idea is to define K centroid one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early group is done. At this point we need to re-calculate k new centroids as centers of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more.
In this method the points are clustered around centroids μ_i $\forall_i$=1,2,3….k which are obtained by minimizing the objective function.

$$\mathbf{V}=\sum_{i=1}^{k}\sum_{xj\,\epsilon\,Si}(xj-\mu i)^2$$

where, there are are k clusters S_i , i=1,2,….k. μ_i is the mean point of all the points $x_j \in S_i$. The algorithm takes 2 dimensional image as input and has various steps that are as follows:
Step 1: Compute the intensity distribution of the intensities.

Step 2: Choose K centroid randomly.
Step 3: Repeat the following steps until the cluster does not change any more.

Step 3.1: Cluster the points based on the distance of their intensities from the centroid intensities.

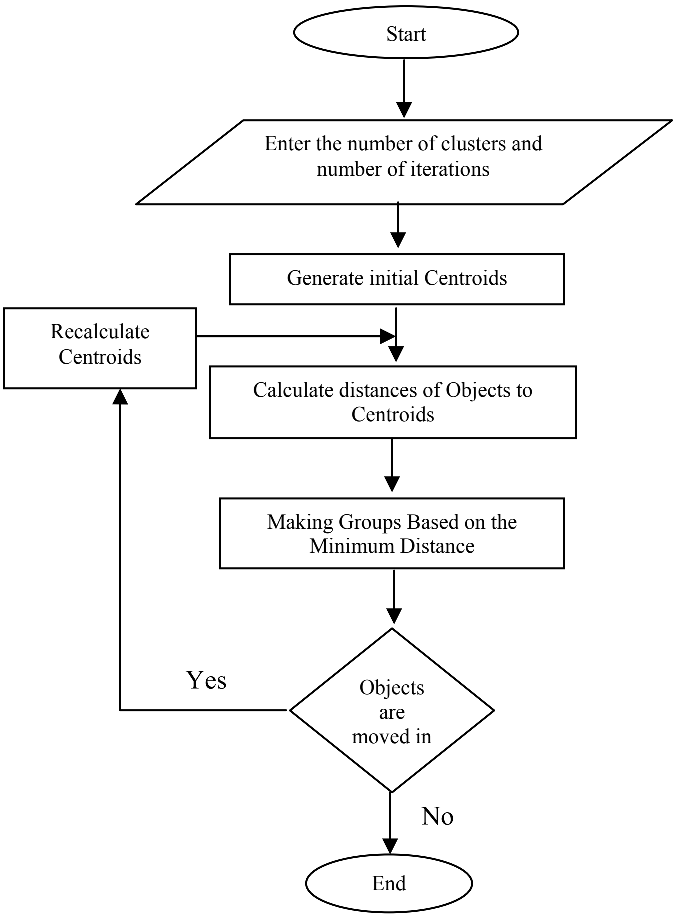
$$c^{(i)} := \arg\min_j \left\| x^i - \mu_j \right\|^2$$

Step 3.2: Compute the new centroid or mean point for each clusters.

$$\mu_i = \frac{\sum_{i=1}^{m} 1\{C(i)=j\}x^{(i)}}{\sum_{i=1}^{m} 1\{C(i)=j\}}$$

Step 4: End

Flow Chart for K-Means Algorithm



III. CANNY OPERATOR

The Canny edge detection operator [5] was developed by John F. Canny in 1986 and uses a multi-stage algorithm to detect a wide range of edges in images. Canny's aim was to discover the optimal edge detection algorithm. From [6] we have studied, in this situation, an "optimal" edge detector means:

- good detection
- good localization
- minimal response

The canny edge detection operator works in the following steps:

Step 1- The first step is to filter out any noise in the original image before trying to locate and detect any edges. In this first we have to smooth the edge with a Gaussian filter to reduce noise and unwanted details and textures.

$$\mathbf{g(m,n) = G\sigma(m.n) * f(m,n)}$$

Step 2- After smoothing the image and eliminating the noise, the next step is to find the edge strength by taking the gradient of the image. The magnitude, or edge strength, of the gradient is then approximated using the formula:

$$|G| = |Gx| + |Gy|$$

Step 3-The direction of the edge is computed using the gradient in the x and y directions. The formula for finding the edge direction is just

Theta = invtan (Gy / Gx)

After the edge direction is known than we have to relate the edge direction to a direction that can be traced in an image. After that we have to apply non-maximum suppression. Non-maximum suppression is used to trace along the edge in the edge direction and suppress any pixel value that is not considered to be an edge. This will give a thin line in the output image.

IV. RMSE

Root-Mean-Square-Error: The RMSE method is a good assessment to find out the accuracy of the transformation between the images. In this the pixels of the original image is denoted by Pi and the pixels of reconstructed image is denoted by Qi (where $1 \leq i \leq n$). For finding out the root mean square error, first we have to find out the mean square error between the two images as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Pi - Qi)^2$$

It is the average of the square of the errors (pixel differences) of the two images. The root mean square error is defined as the square root of the mean square error method.

$$RMSE = \sqrt{MSE}$$

V. RESULT

Here we are using different images and then apply canny operator directly on the image as well as apply the canny operator on the image after segmenting it. Figure 5.1.1 is the original image of lena and when we apply canny operator on that image then we get figure 5.1.2, it has been observed that the output is very clumsy and contains a lots discontinuous edges and because of this region in an image is not clearly specified. Because of this reason we are segmenting the original image by using K-Means clustering algorithm. Figure 5.1.3 shows the output when we segment the image into two segments and after that apply the canny operator on the segmented image, then the output shows continuous and smooth edges but edges with closed boundaries is very less, it means regions are not clear. But as we increases the number of segments the continuity between the edges increases as well as we are getting the more number of regions with closed boundaries and extracting a very small region is also possible for further enhancement.

Figure 5.1.1(Original Image)

Figure 5.1.2(Canny Operator)

Figure 5.1.3(Canny Operator After 2 Segments)

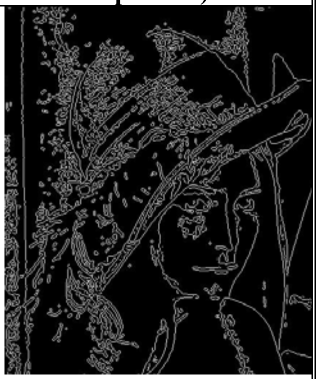


Figure 5.1.4(Canny Operator After 4 Segments)

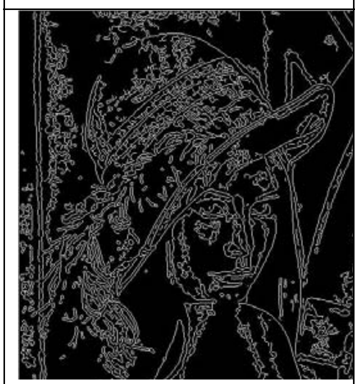


Figure 5.1.5(Canny Operator After 8 Segments)

Figure 5.1.6(Canny Operator After 16 Segments)

Figure 5.2.1(Original Image)

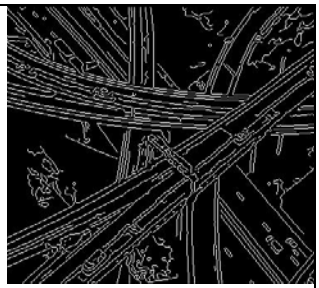


Figure 5.2.2(Canny Operator)

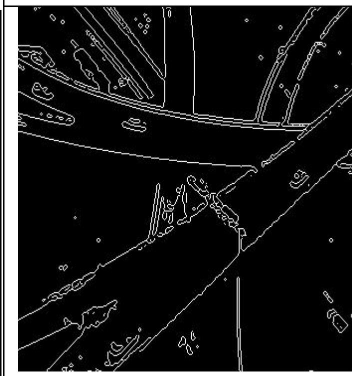


Figure 5.2.3(Canny Operator After 2 Segments)

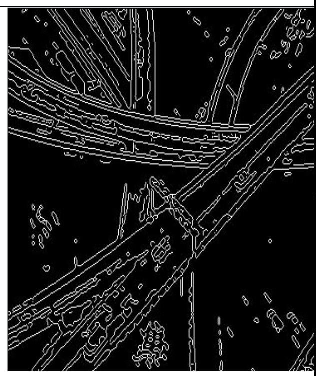


Figure 5.2.4(Canny Operator After 4 Segments)

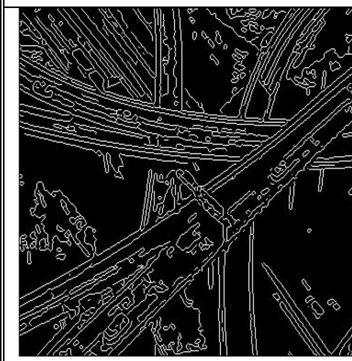


Figure 5.2.5(Canny Operator After 8 Segments)

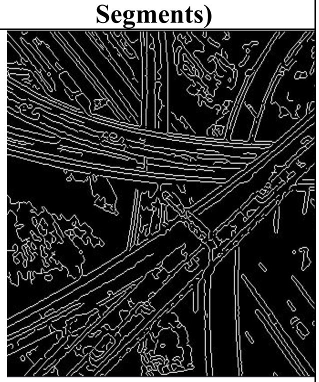


Figure 5.2.6(Canny Operator After 16 Segments)

Figure 5.3.1(Original Image)

Figure 5.3.2(Canny Operator)

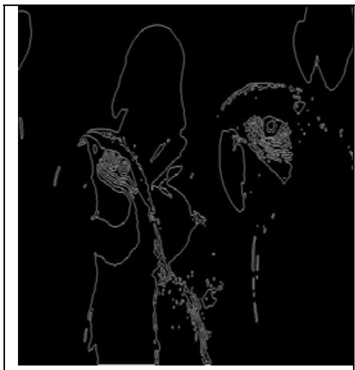


Figure 5.3.3(Canny Operator After 2 Segments)

Figure 5.3.4(Canny Operator After 4 Segments)

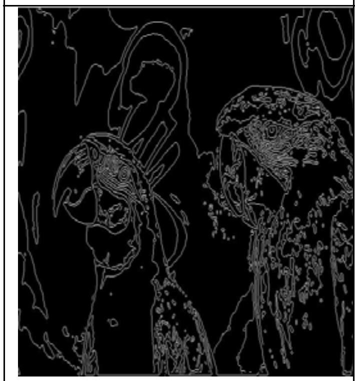


Figure 5.3.5(Canny Operator After 8 Segments)

Figure 5.3.6(Canny Operator After 16 Segments)

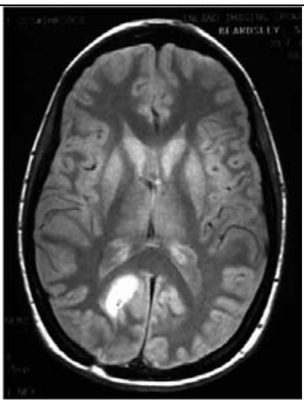


Figure 5.4.1(Original Image)

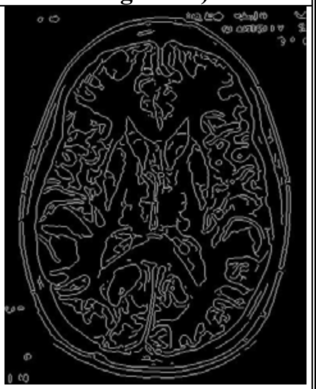


Figure 5.4.2(Canny Operator)

Figure 5.4.3(Canny Operator After 2 Segments)

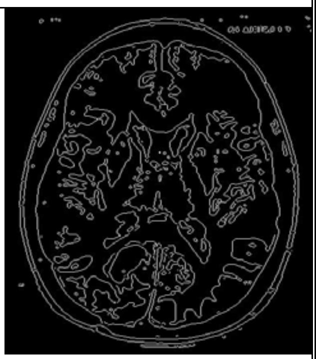


Figure 5.4.4(Canny Operator After 4 Segments)

Figure 5.4.5(Canny Operator After 8 Segments)

Figure 5.4.6(Canny Operator After 16 Segments)

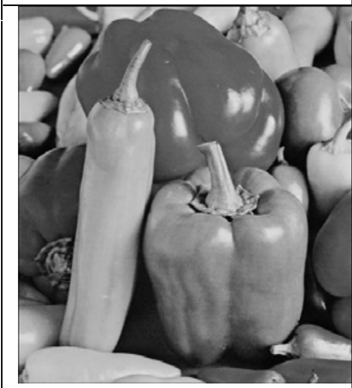


Figure 5.5.1(Original Image)

Figure 5.5.2(Canny Operator)

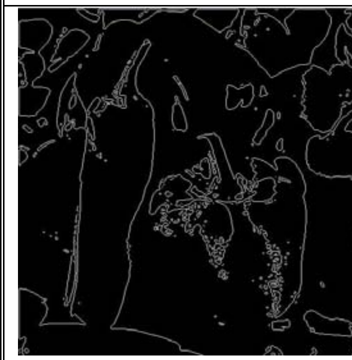


Figure 5.5.3(Canny Operator After 2 Segments)

Figure 5.5.4(Canny Operator After 4 Segments)

Figure 5.5.5(Canny Operator After 8 Segments)

Figure 5.5.6(Canny Operator After 16 Segments)

Figure 5.1 RMSE Between the Output of Original Canny and the Output of the Canny Applied On Segmented Image

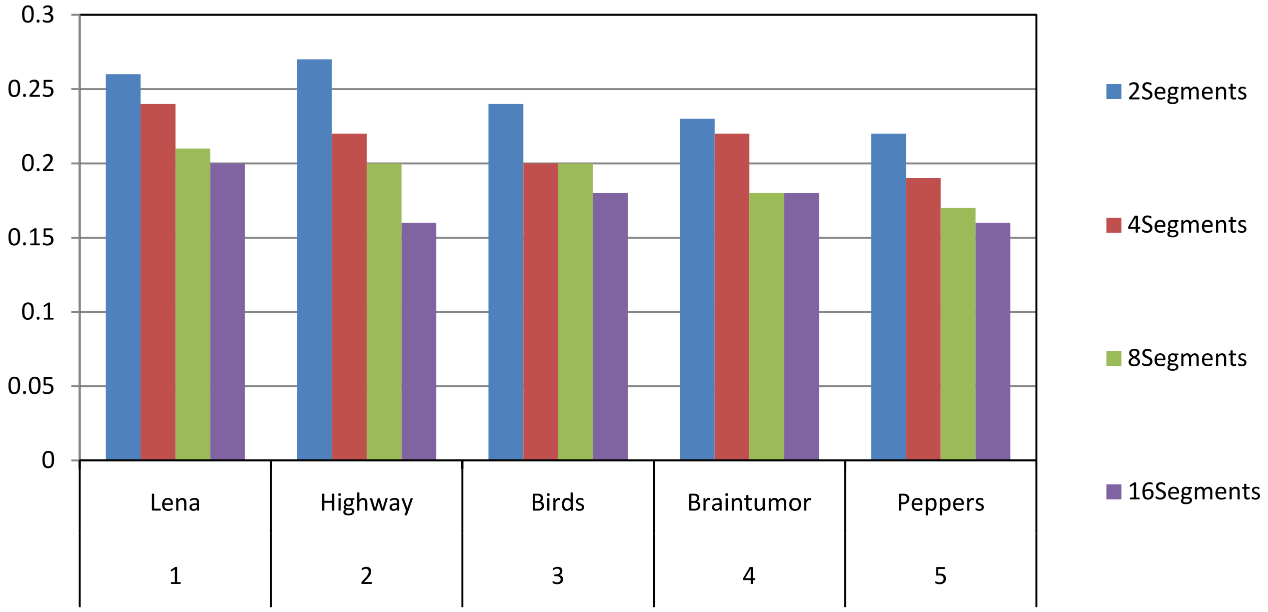


Table 5.1 RMSE between the Output of Original Canny and the Output of the Canny Applied On Segmented Image

Image	2Segments	4Segments	8Segments	16Segments
Lena	0.26	0.24	0.21	0.2
Highway	0.27	0.22	0.2	0.16
Birds	0.24	0.2	0.2	0.18
Braintumor	0.23	0.22	0.18	0.18
Peppers	0.22	0.19	0.17	0.16

VI. CONCLUSION AND FUTURE WORK

As we have seen in figure 5.1.2, that if we apply canny operator directly on any image, the result shows a lot of discontinuous edges but if we segment the image before detection of edges, it produces quite smoother and continuous edges. In this paper we used k-Means clustering algorithm to segment the image and after that we apply the canny operator on the segmented image. Table 5.1 shows the root mean square error value between the output generated by the canny operator directly and the output generated after segmenting the image and then apply canny operator on it. As the number of segments increases the value of RMSE decreases and when its value decreases we are getting the output image that contain quite smooth and continuous edges. So now, we can say that by segmenting the image into number of segments given by the user as per the requirement for the application and then apply canny operator on the segmented image gives better result.

Our future work is to club various numbers of pixels into one cluster and the number of pixel to be clubbed is given by the user on experimental basis, to get much better result that helps us in the enhancement of images.

REFERENCES

[1] N. SenthilKumaran, & R.Rajesh (May 2010). A Note on Image Segmentation Techniques. *International Journal of Recent Trends in Engineering and Technology, Vol 3,No.2* , 3.

[2] N. SenthilKumaran, R.Rajesh (May 2009). Edge Detection Techniques for Image Segmentation. *International Journal of Recent Trends in Engineering, Vol.1, No.2* , 5.

[3] Wen-Xiong kang, Qing-Qang Yang, Run-Peng Liang. The Comparative Research on Image Segmentation Algorithms. 5.

[4] Y. Ramadevi, T.Sridevi,B. Poornima & B. Kalyani(December 2010). Segmentation and Object Recognition Using Edge Detection Techniques. *International Journal of Computer science and Information Technology, Vol.2, No.6* , 9.

[5] Megha Seth, Dr. Sipi Dubey (February 2012). An Analytic Approach of Edge Detection Gradient Operator. *International Journal Computer Technology and Applications, Vol.3* , 4.

[6] Osama Abu Abbas, (July 2008). Comparison Between Data Clustering Algorithms. *The International Arab Journal of Information Technology, Vol.5, No.3* , 6.