Performance and Analysis of Image Filtering Techniques

Shradha Dakhare#, Harshal Chowhan*

*Computer Science Department
W.C.E.M, Nagpur, India
1srode.dakhare@gmail.com
2chowhan23@gmail.com

ABSTRACT:
Many image segmentation techniques have been developed over the past two decades for segmenting the images, which help for object recognition, occlusion boundary estimation within motion or stereo systems, image compression, image editing. The objective is to develop a combined approach for segmenting the image, which give better segmented output image. First we used histogram equalization to the input image, from which we get contrast enhancement output image. After that we applied median filtering which will remove noise from contrast output image. At last we applied fuzzy c-mean clustering algorithm to denoising output image, which give segmented output image. In this paper, we compare the performance of various segmentation techniques i.e. histogram equalization, median filter and fuzzy c means.

Keywords: Histogram equalization, Median filter, Fuzzy C-Means

I. INTRODUCTION
Image segmentation refers to the major step in image Processing in which the inputs are images and, outputs are the attributes extracted from those images. It is an important and challenging problem in image analysis as well as in high-level image interpretation and understanding such as robot vision, object recognition, and medical imaging. Segmentation divides image into its constituent Regions or objects. The level to which segmentation is carried out depends upon the problem being solved i.e. segmentation should stop when the objects of interest in an application have been isolated. Image segmentation refers to the decomposition of a scene into its components [1].

Quantitative studies have been performed based on populations of biological Images. Such studies extremely require methods for segmentation, feature extraction, and classification. A first step in many analysis pipelines is segmentation, which can occur at several levels (e.g., separating nuclei, cells, tissues). This task has been an active field of research in image processing over the last 30 years, the modality, quality, and resolution of the Microscopy images to analyze[2].

II. HISTOGRAM EQUALIZATION

Histogram equalization is a kind of contrast enhancement that stretches the histogram so that all values occur (more or less) an equal number of times. This often works well, especially for bringing out details in overly light or overly dark sections of a gray scale image. The transformation is scaled such that the least intense value in the original image is mapped to a zero intensity value in the equalized image. As well, the most intense value in the original image is mapped to an intensity value that is equal to the maximum intensity value determined by the bit depth of the image. This produces results that have a dynamic range that is slightly larger than produced by the histogram equalization algorithm described in Gonzalez and Woods (2008). Adaptive histogram equalization (AHE) is a computer image processing technique used to improve contrast in images. It differs from ordinary histogram in the respect that the adaptive method computes several histogram, each corresponding to a distinct section of the image, and uses them to redistribute the lightness values of the image. It is therefore suitable for improving the local contrast of an image and bringing out more detail.

The algorithm has been tested to verify that performing further equalization on an already equalized image produces an output that is identical to the input. As well, it has been tested to verify that the histogram equalization of an input image with a constant probability distribution function produces an output image that is identical to the input. The MatLab script used to run these tests is provided in
the Tests.m file [3]original image on the left and the 
equilized image on the right.

A. Histogram Equalization Method
Consider a discrete grayscale image \( \{ x \} \) and 
let \( n_i \) be the number of occurrences of gray level \( i \). 
The probability of an occurrence of a pixel of level \( i \) 
in the image is

\[
p_x(i) = p(x = i) = \frac{n_i}{n}, \quad 0 \leq i < L
\]

\( L \) being the total number of gray levels in the image, 
\( n \) being the total number of pixels in the image, and 
\( p_x(i) \) being in fact the image histogram for pixel value \( i \), normalized to \([0,1]\). The cumulative 
distribution function corresponding to \( p_x \), as

\[
cdf_x(i) = \sum_{j=0}^{i} p_x(j)
\]

Which is also the image’s accumulated normalized 
histogram? We would like to create a transformation 
of the form \( y = T(x) \) to produce a new image \( \{ y \} \), such 
that its CDF will be linearized across the value range, 
i.e.

\[
y = T(x) = cdf_x(x)
\]

\[
cdf_y(i) = iK
\]

For some constant \( K \). The properties of the CDF 
allow us to perform such a transform it is defined as, 
Notice that the \( T \) maps the levels into the range \([0, 1] \). 
In order to map the values back into their original range, the following simple transformation needs to 
be applied on the result:

\[
y' = y \cdot (\max\{x\} - \min\{x\}) + \min\{x\}
\]

The above describes histogram equalization on a 
grayscale image [4]. However it can also be used on 
color images by applying the same method separately 
to the Red, Green and Blue components of the RGB 
color values of the image. However, applying the 
same method on the Red, Green, and Blue 
components of an RGB image may yield dramatic 
changes in the image’s color balance since the relative 
distributions of the color channels change as a result 
of applying the algorithm. However, if the image is 
first converted to another color space, Lab color 
space, or HSL/HSV color space in particular, then the 
algorithm can be applied to the luminance or value 
channel without resulting in changes to the hue and 
saturation of the image [5]. There are several 
histogram equalization methods in 3D space.

Trahanias and Venetsanopoulos applied histogram 
equalization in 3D color space[6].

I. MEDIAN FILTER
The median filter is a nonlinear digital filtering 
technique, often used to remove noise. To reduce 
noise several non-linear filters can be employed. One 
of the simplest techniques, the median filter, provided 
good noise reduction without affecting the borders of 
the objects on the image. The main idea of the median 
filter is to run through the signal entry by entry, 
replacing each entry with the median of neighboring 
entries. The pattern of neighbors is called the 
"window", which slides, entry by entry, over the 
entire signal.

Consider median filter with three point window of 
\( W = \{-1, 0, 1\} \) and the inputs are \( \{x(n-1), x(n), x(n+1)\} = \{-3, -6, 8\} \). To formulate the function [9]

\[
F(\theta) = \sum | \theta - x_{n+k} | 
k= -1
\]

Often though, at the same time as reducing the noise 
in a signal, it is important to preserve the edges. 
Edges are of critical importance to the visual 
appearance of images, for example. For small to 
moderate levels of (Gaussian) noise, the median filter 
is demonstrably better than Gaussian blur at removing 
noise whilst preserving edges for a given, fixed 
window size[9]. However, its performance is not that 
much better than Gaussian blur for high levels of 
noise, whereas, for speckle noise[12](impulsive noise), 
it is particularly effective[10]. Because of this, 
median filtering is very widely used in digital image 
processing.

III. FUZZY C-MEANS CLUSTERING
The fuzzy segmentation methods, which can 
retain more information from the original than hard 
segmentation methods [11-12]. In hard clustering, data 
is divided into distinct clusters, where each data 
element belongs to exactly one cluster. In fuzzy 
clustering (also referred to as soft clustering), data 
elements can belong to more than one cluster, and 
associated with each element is a set of membership 
levels. These indicate the strength of the association 
between that data element and a particular cluster. 
Fuzzy clustering is a process of assigning these 
membership levels, and then using them to assign 
data elements to one or more clusters [13].

One of the most widely used fuzzy clustering 
algorithm is the Fuzzy C-Means (FCM) Algorithm 
(Bezdek 1981). The FCM algorithm attempts to 
partition a finite collection of \( n \) elements \( X = \{x_1, ..., x_n\} \) into a collection of \( c \) fuzzy clusters with 
respect to some given criterion. Given a finite set of
data, the algorithm returns a list of $c$ cluster centers $C = \{c_1, c_2, \ldots, c_c\}$ and a partition.

$$J=U = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}$$

Where each element $u_{ij}$ tells the degree to which element $x_i$ belongs to cluster $c_j$.

In fuzzy clustering, each point has a degree of belonging to clusters, as in fuzzy logic, rather than belonging completely to just one cluster. Thus, points on the edge of a cluster may be in the cluster to a lesser degree than points in the center of cluster. Any point $x$ has a set of coefficients giving the degree of being in the $k$th cluster $w_k(x)$. With fuzzy $c$-means, the centroid of a cluster is the mean of all points, weighted by their degree of belonging to the cluster:

$$C_k = \frac{\sum x w_k(x)}{\sum w_k(x)}$$

The degree of belonging, $w_k(x)$, is related inversely to the distance from $x$ to the cluster center as calculated on the previous pass. It also depends on a parameter $m$ that controls how much weight is given to the closest center[14].

A. The algorithm of fuzzy c-means clustering

Step1. Choose a number of clusters in a given image.
Step2. Assign randomly to each point coefficients for being in a cluster.
Step3. Repeat until convergence criterion is met.
Step4. Compute the center of each cluster.
Step5. For each point, compute its coefficients of being in the cluster [4-5].

Experiment No 1:

<table>
<thead>
<tr>
<th>TIME</th>
<th>MSE</th>
<th>PSNR</th>
<th>DATA LOSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>N H</td>
<td>2.49s</td>
<td>0.015dB</td>
<td>84.79dB</td>
</tr>
<tr>
<td>AH</td>
<td>1.70s</td>
<td>0.005dB</td>
<td>93.59dB</td>
</tr>
</tbody>
</table>

Experiment No 2:

<table>
<thead>
<tr>
<th>TIME</th>
<th>MSE</th>
<th>PSNR</th>
<th>DATA LOSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MF</td>
<td>0.54s</td>
<td>0.055dB</td>
<td>73.42dB</td>
</tr>
</tbody>
</table>

Experiment No 3:
TABLE III

<table>
<thead>
<tr>
<th>TIME</th>
<th>MSE</th>
<th>PSNR</th>
<th>DATA LOSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCM</td>
<td>87.80s</td>
<td>0.55dB</td>
<td>53.32dB</td>
</tr>
</tbody>
</table>

TABLE IV

<table>
<thead>
<tr>
<th>TIME</th>
<th>MSE</th>
<th>PSNR</th>
<th>DATA LOSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>HE</td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MF</td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FCM</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
</tbody>
</table>

V. CONCLUSIONS

This paper presented a comparison study between histogram equalization, median filter and fuzzy c means. Future research will concentrate on the combined approach that is histogram equalization, median filtering, and fuzzy c-means algorithm for image segmentation which gives the better segmentation with reduced computational time.

REFERENCES

[4] Histogram equalization –wikipedia, the free encyclopedia http://en.wikipedia.org/wiki/Histogram_equalization#cite_ref-