*International Journal For Technological Research In Engineering*
*Volume 1, Issue 8, April-2014* *ISSN (Online): 2347 - 4718*

# A HYBRID THRESHOLDING APPROACH FOR IMPROVING THE QUALITY OF BINARY IMAGES

Vibhoo Sharma[1], Noor Mohammed[2]
[1]Research Scholar, [2]Assistant Professor

***Abstract: -Recent years have witnessed the rapid growth of degraded images due to the increasing power of Computing and the fast development of Internet. Because of this tremendous increase of quality of degraded images, there is an urgent need of image Content description to facilitate automatic retrieval. Image is described by several low level image features, such as color, texture, shape or the combination of these features. Shape is an important low level image feature. In this thesis, enhancement of quality of degraded image using global Thresholding This thesis is primarily concerned with the extracting numerical features from binary images. Image processing is very vast field & one of the most important parts of image processing is thresholding. Thresholding, which is an important pre-processing steps for the degraded image to enhance their quality & has been studies in relation to various images. There are different algorithms that have been used & studies for various factors. Image analysis Value of Thresholding based on which segmentation has been performed.***

## I. PROBLEM DEFINITION AND OBJECTIVES

It is common for libraries to provide public access to historical and ancient document image collections. It is common for such document images to require specialized processing in order to remove background noise and become more legible. In this paper, we propose a hybrid binarizatin approach for improving the quality of old documents using a combination of global and local thresholding. First, a global thresholding technique specifically designed for old document images is applied to the entire image. Then, the image areas that still contain background noise are detected and the same technique is reapplied to each area separately. Hence, we achieve better adaptability of the algorithm in cases where various kinds of noise coexist in different areas of the same image while avoiding the computational and time cost of applying a local thresholding in the entire image. Evaluation results based on a collection of historical document images indicate that the proposed approach is effective in removing background noise and improving the quality of degraded documents while documents already in good condition are not affected. Recognition/classification of objects and patterns independent of their position, size, orientation and other variations in geometry and colors has been the goal of much recent research. Image thresholding converts the gray level image in to binary one. The two binary levels may represent objects and background or more generally to classes in an image. Pixels whose value exceeds the critical value are assignees to one category & the rest to other. The thresholding is global if the same critical value is used over the whole images. Many algorithms have been proposed for selecting the threshold appropriate for the given image. Some algorithms only used the histogram of value in the image. That is the no of pixels at each gray level Global histodiagram based algorithms are the most commonly used despite the benefits that can accrue from using contextual information & allowing the threshold to vary over an image. They can simple understand & implement. Several groups of features have been used for this purpose, such as simple visual features (edges, contours, textures, etc.). The interpretation of images is still a complex problem and is primordial in lots of applications (spatial, military, industrial and medical). Since the beginning of the 80's, lots of research works have been achieved for the conception of vision systems in order to recognize objects in an image.

An essential stage concerns the strategy of object recognition because an object can appear at different places in the image or at different orientations and scales. Document images as a substitute for paper document main. It consists of common symbols as hand written or machine printed characters, symbols & graphics.

## II. GREY-LEVEL IMAGES

A 3D image is generally grey-level and hard to transform into a bi-level image without loss of important information. We are interested in finding a a shape representation, where the grey-level distribution of the image is in focus. As volume images contain large amounts of data, but the information relevant for shape analysis is actually carried by a limited number of voxels, there is an interest for representation schemes with reduced dimensionality, e.g., the skeleton. Representing the foreground in terms of entities of lower dimensionality, surfaces and curves, is not a straightforward task, as the relevant information is not evenly distributed. At the moment we refer to image domains where the relevant information is gathering in correspondence with the foreground subsets with locally higher intensities, so that surfaces and curves should be found mainly therein. This is the case, e.g., for MRA

(Magnetic Resonance Angiography) images, where the foreground is the blood vessels, which are characterized by high intensities [25]. We use grey-level and distance information to reduce the foreground of a grey-level 3D image to a medial surface representation. This representation is topologically equivalent to the initial foreground and is mainly constituted by surfaces (and curves) centered within regions having locally higher intensities. In a grey-level image with grey-levels $g_0$ to $g_n$, voxels with grey-level $g_0$ are considered as background and the remaining voxels are foreground. In the 3D example below, grey-levels are represented by colures with increasing hue angle from red=$g_1$ to blue=$g$

## III. STUDY OF CONTOUR BASED SHAPE DESCRIPTORS

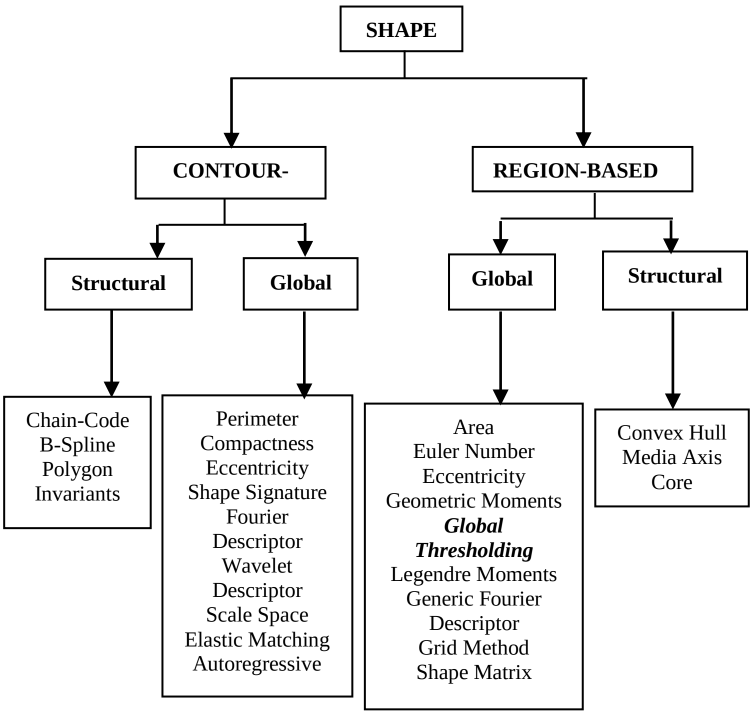



Figure 1.1. Taxonomy of shape description techniques.

## IV. SEGMENTATION

Segmentation is very important to image retrieval. Both the shape feature and the layout feature depend on good segmentation. In this subsection we will describe some existing segmentation techniques used in both computer vision and image retrieval. Lybanon et al. researched the morphological operation (opening and closing) based approach in image segmentation. They tested their approach in various types of images, including optical astronomical images, infrared ocean images, and magnetograms. While this approach was effective in dealing with the above scientific image types, its performance needs to be further evaluated for more complex natural scene images. Hansen and Higgins exploited the individual strengths of watershed analysis and relaxation labeling. Since a fast algorithm exists for watershed, they first used watershed to subdivide an image into catchmen basins. They then used relaxation labeling to refine and update the classification of catchmen basins initially obtained from watershed to take advantages of relaxation labeling robustness to noise. Li et al. proposed a fuzzy-entropy-based segmentation approach [23]. This approach is based on the fact that local entropy maxima correspond to the uncertainties among various regions in the image. This approach was very effective for images whose histogram do not have clear peaks and valleys. Other segmentation techniques based on Delaunay triangulation are fractals, and edge flow [12]. All the above mentioned algorithms are automatic. A major advantage of this type of segmentation algorithm is that it can extract boundaries from large numbers of images with out occupying the user's time and effort. However, in an unconstrained domain, for non-preconditioned images, the automatic segmentation is not always reliable. What an algorithm can segment in this case is only regions, not objects. To obtain high-level objects, which is desirable in image retrieval, human assistance is needed. Samadani and Han proposed a computer-assisted boundary extraction approach, which combined manual inputs from the user with the image edges generated by the computer. Daneels et al. developed an improved method of active contours. Based on the user's input, the algorithm first used a greedy procedure to provide fast initial convergence. Secondly, the outline was refined by using dynamic programming [23]. Rui et al. proposed a segmentation algorithm based on clustering and grouping in spatial-color-texture space. The user defines where the attractor (object of interest) is, and the algorithm groups regions into meaningful objects. Last comment worth mentioning in segmentation is that the requirements of segmentation accuracy are quite different for shape features and layout features. For the former, accurate segmentation is highly desirable, whereas for the latter, a coarse segmentation suffices. As we can see from the above descriptions, many visual features have been explored, both previously in computer vision applications and currently in image retrieval applications. For each visual feature, there exist multiple representations, which model the human perception of that feature from different perspectives. What features and representations should be used in image retrieval depends on the application. There is a need to develop an image-content description (model) to organize the features. The features not only should be associated with the images but also should be invoked at the right place and time, whenever they are needed to assist retrieval. A histogram of the input image intensity should reveal two peaks corresponding respectively to the signals from the background and the object. The global;

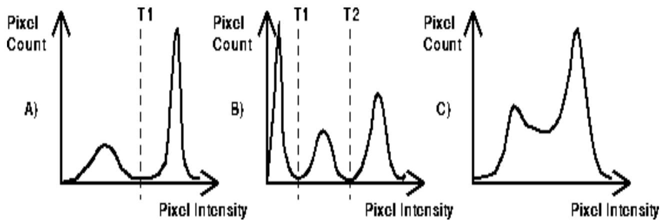
  

thresholding option in 3DMA allows the answer the user to pick a single global threshold for a 3D image of separate threshold for each 2D slice in the image. Some experimental option has also been provided to automatic choice of threshold by performing a bi-normal fit to the two peak histogram and setting a Threshold at the inerpeak minimum as determined by the normal fits. The thresholding option output the segmented image slice wise. In a packed bit (0, 1) format. All voxels having intensity below the thresholding value are set to 0, the rest are set to 1. 'Washy Talky' washing machine from Electrolux-Kelvinator would be an ideal choice for an upper-class consumers. However, the same wouldn't be of much use to a grandmother. Consumers buy a product taking into account their preferences. The preferences change according the financial background, education, and experience [6]. The differences in the consumer preferences and behavioral patterns pose a challenge to Abhinash Patnaik, who intends to open an Electrolux showroom in Bhubaneswar. He's confused as to how he should differentiate his offerings to suit the taste of every customer. Companies, whether they are big or small experience a similar challenge. In such a confused situation, 'Segmentation' comes in handy! Segmentation is defined as the process of dividing a market into distinct sub-sets of consumers with common needs or characteristics and selecting one or more segments to target. Segmentation is the first step in a marketing strategy. Once marketers divide the market into various groups, they can then select their 'targeted segments' and design products that suit their requirements. For instance, companies like BPL have resorted to market segmentation as a strategy to beat its competitors. Its products like: BPL Loews (digital designer televisions for the premium-end of the market); Matrix Flat screen TV (for technology lovers); Studio line (who are performance seekers); and Prima (for the lower-end of the market) are a case in point. Each of them has been designed to cater to the requirement of a particular segment.

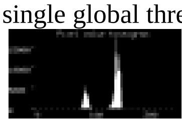
In many vision applications, it is useful to be able to separate out the regions of the image corresponding to objects in which we are interested, from the regions of the image that correspond to background. Thresholding often provides an easy and convenient way to perform this segmentation on the basis of the different intensities or colours in the foreground and background regions of an image. In addition, it is often useful to be able to see what areas of an image consist of pixels whose values lie within a specified range, or band of intensities (or colours). Thresholding can be used for this as well [23]. The input to a thresholding operation is typically a grayscale or colour image. In the simplest implementation, the output is a binary image representing the segmentation. Black pixels correspond to background and white pixels correspond to foreground (or vice versa). In simple implementations, the segmentation is determined by a single parameter known as the intensity threshold. In a single pass, each pixel in the image is compared with this threshold. If the pixel's intensity is higher than the threshold, the pixel is set to, say, white, in the output. If it is less than the threshold, it is set to black. In more sophisticated implementations, multiple thresholds can be specified, so that a band of intensity values can be set to white while everything else is set to black. For colour or multi-spectral images, it may be possible to set different thresholds for each colour channel, and so select just those pixels within specified cuboids in RGB space. Another common variant is to set to black all those pixels corresponding to background, but leave foreground pixels at their original colour/intensity (as opposed to forcing them to white), so that that information is not lost[12].

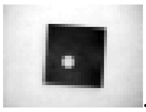
Not all images can be neatly segmented into foreground and background using simple thresholding. Whether or not an image can be correctly segmented this way can be determined by looking at an intensity histogram of the image. We will consider just a grayscale histogram here, but the extension to colour is trivial [15]. If it is possible to separate out the foreground of an image on the basis of pixel intensity, then the intensity of pixels within foreground objects must be distinctly different from the intensity of pixels within the background. In this case, we expect to see a distinct peak in the histogram corresponding to foreground objects such that thresholds can be chosen to isolate this peak accordingly. If such a peak does not exist, then it is unlikely that simple thresholding will produce a good segmentation. In this case, adaptive thresholding may be a better answer [23].

Figure 1 shows some typical histograms along with suitable choices of threshold.

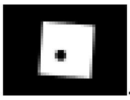
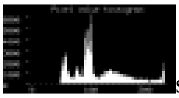
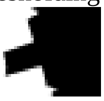
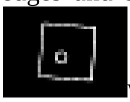
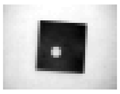
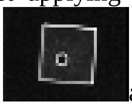
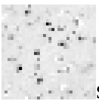
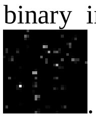
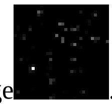
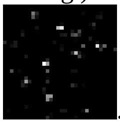



single global threshold

shows the histogram for the image

. This shows a nice bi-modal distribution --- the lower peak represents the object and the higher one represents the background. The picture can be segmented using a single threshold at a pixel intensity value of 120.

The result is shown in .
shows the histogram for the image . Due to the severe illumination gradient across the scene, the peaks corresponding to foreground and background have run together and so simple thresholding does not give good results.  and  show the resulting bad segmentations for single threshold values of 80 and 120 respectively. (Reasonable results can be achieved by using adaptive thresholding on this image.)
Thresholding is also used to filter the output of or input to other operators[14]. For instance, in the former case, an edge detector like Sobel will highlight regions of the image that have high spatial gradients. If we are only interested in gradients above a certain value (i.e. sharp edges), then thresholding can be used to just select out the strongest edges and set everything else to black. As an example, was obtained by first applying the Sobel operator to  to produce and then thresholding this using a threshold value of 60.
Thresholding can be used as preprocessing to extract an interesting subset of image structures which will then be passed along to another operator in an image processing chain. For example, image shows a slice of brain tissue containing nervous cells (i.e. the large grey blobs, with darker circular nuclei in the middle) and glia cells (i.e. the isolated, small, black circles). We can threshold this image so as to map all pixel values between 0 and 150 in the original image to foreground (i.e. 255) values in the binary image, and leave the rest to go to background . The resultant image can then be connected components labeled in order to count the total number of cells in the original image. If we wanted to know how many nerve cells there are in the original image, we might try applying a double threshold in order to select out just the pixels which correspond to nerve cells (and therefore have middle level grayscale intensities) in the original image. (In remote sensing and medical terminology, such thresholding is usually called density slicing.) Applying a threshold band of 130 - 150 yields . While most of the foreground of the resulting image corresponds to nerve cells, the foreground features are so disconnected (because nerve cell nuclei map to background intensity values along with the glia cells) that we cannot apply connected components labeling. Alternatively, we might obtain a better assessment of the number of nerve cells by investigating some attributes (. size, as measured by a distance transform) of the binary image containing both whole nerve cells and glia. In reality, sophisticated modeling and/or pattern matching is required to segment
An alternative approach to finding the local threshold is to statistically examine the intensity values of the local neighborhood of each pixel. The statistic which is most appropriate depends largely on the input image. Simple and fast functions include the mean of the local intensity distribution,

$$T = mean$$

the median value,

$$T = median$$

or the mean of the minimum and maximum values,

$$T = \frac{max - min}{2}$$

The size of the neighborhood has to be large enough to cover sufficient foreground and background pixels, otherwise a poor threshold is chosen. On the other hand, choosing regions which are too large can violate the assumption of approximately uniform illumination. This method is less computationally intensive than the Chow and Kanenko approach and produces good results for some applications.

Like global thresholding, adaptive thresholding is used to separate desirable foreground image objects from the background based on the difference in pixel intensities of each region. Global thresholding uses a fixed threshold for all pixels in the image and therefore works only if the intensity histogram of the input image contains neatly separated peaks corresponding to the desired subject(s) and background(s). Hence, it cannot deal with images containing, for example, a strong illumination gradient.

Local adaptive thresholding, on the other hand, selects an individual threshold for each pixel based on the range of intensity values in its local neighborhood. This allows for thresholding of an image whose global intensity histogram doesn't contain distinctive peaks. A task well suited to local adaptive thresholding is in segmenting text from the

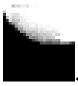
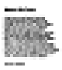
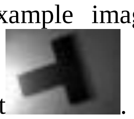
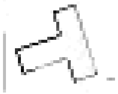
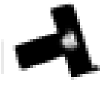
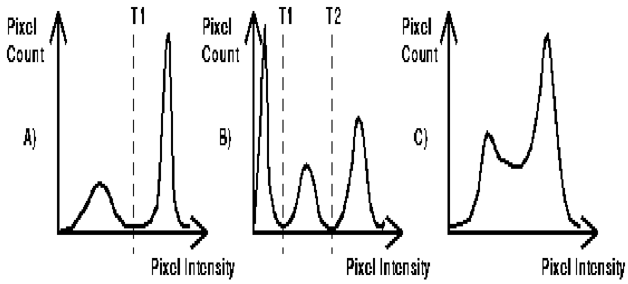
image. Because this image contains a strong illumination gradient, global thresholding produces a very poor result, as can be seen in .
Using the mean of a 7×7 neighborhood, adaptive thresholding yields. The method succeeds in the area surrounding the text because there are enough foreground and background pixels in the local neighborhood of each pixel - i.e. the mean value lies between the intensity values of foreground and background and, therefore, separates easily. On the margin, however, the mean of the local area is not suitable as a threshold, because the range of intensity values within a local neighborhood is very small and their mean is close to value of the center pixel.
The situation can be improved if the threshold employed is not the mean, but (mean-C), where C is a constant. Using this statistic, all pixels which exist in a uniform neighborhood ( along the margins) are set to background. The result for a 7×7 neighborhood and C = 7 is shown in  and for a 75×75 neighborhood and C = 10 in . The larger window yields the poorer result, because it is more adversely affected by the illumination gradient. (Also note that the latter is more computationally intensive than thresholding using the smaller window.
The result of using the median instead of the mean can be seen in . (The neighbourhood size for this example is 7×7 and C = 4. The result shows that, in this application, the median is a less suitable statistic than the mean.
Consider another example image containing a strong illumination gradient. This image cannot be segmented with a global threshold, as shown in , where a threshold of 80 was tried. However, since the image contains a large object, it is hard to apply adaptive thresholding, as well. Using the (mean-C) as a local threshold, we obtain  with a 7×7 window and C = 4, and  with a 140×140 window and C = 8. All pixels which belong to the object but do not have any background pixels in their
Neighborhoods are set to background. The latter image shows a much better result than that achieved with a global threshold, but it is still missing some pixels in the center of the object. In many applications, computing the mean of a neighborhood (for each pixel!) whose size is of the order 140×140 may take too much time. In this case, the more complex Chow and Kanenko approach to adaptive thresholding would be more successful. Brief Description In many vision applications, it is useful to be able to separate out the regions of the image corresponding to objects in which we are interested, from the regions of the image that correspond to background. Thresholding often provides an easy and convenient way to perform this segmentation on the basis of the different intensities or colours in the foreground and background regions of an image. In addition, it is often useful to be able to see what areas of an image consist of pixels whose values lie within a specified range, or band of intensities (or colours). Thresholding can be used for this as well [12]. The input to a thresholding operation is typically a grayscale or colour image. In the simplest implementation, the output is a binary image representing the segmentation [10]. Black pixels correspond to background and white pixels correspond to foreground (or vice versa). In simple implementations, the segmentation is determined by a single parameter known as the intensity threshold. In a single pass, each pixel in the image is compared with this threshold. If the pixel's intensity is higher than the threshold, the pixel is set to, say, white, in the output. If it is less than the threshold, it is set to black[24].
In more sophisticated implementations, multiple thresholds can be specified, so that a band of intensity values can be set to white while everything else is set to black. For colour or multi-spectral images, it may be possible to set different thresholds for each colour channel, and so select just those pixels within specified cuboids in RGB space. Another common variant is to set to black all those pixels corresponding to background, but leave foreground pixels at their original colour/intensity (as opposed to forcing them to white), so that that information is not lost.

Figure 1 shows some typical histograms along with suitable choices of threshold.



## V. DESIGN & ARCHITECTURE

*5.1 Intermean Method*
Let I = gray level (0-255)
& yi = no. of pixels with gray level I
= height of level I in the histogram.
Three partial sums are calculated as:
Three partial sums are calculated as:

$$Ai \quad \sum_{I=O}^{j} yi$$

$$Bj \quad \sum_{I=O}^{j} (i * yi)$$

The mean gray levels in the two classes defined by the threshold below & above it are calculated as:

$$Cj \quad \sum_{I=0}^{j} (i * I * yi)$$

In this method t can be calculated as an integral part of Bn/An. This is an initial guess for t.

$$\mu t \quad Bt / At$$

$$\sqrt{t} = \frac{(Bn - Bt)}{(An - At)}$$

$$\frac{(\mu t - \sqrt{t})}{2}$$

Then & are recalculated & a new value of t is obtained. This process is repeated until convergence.

It is calculated as: - $\sqrt{\ }\mu$

*5.2 Minerror method*
In this method, histogram can be viewed as an estimate of the probability density function p(g) of the mixture of the population comprising the gray levels of objects & background pixels.

P (g/I) = prob. Density function
Two components 1 – 1, 2
$\mu i$ = mean of mixture

$\partial I$ = standard deviation

pi= a priori probability
Supposing gray level data is thresholding at T then:

$$Pi(T) = \sum_{g=a}^{b} h(g)$$

$$\mu i(T) = \sum_{g=a}^{b} g * h(g) / pi(T)$$

$$\&\partial i \,\&\, \partial i(T) = \left| \sum \{g - \mu i(T) * g - \mu i(T)\} * h(g) \right| / pi(T)$$

Where a = 0 i=0
T+1 i = 2
b = T i –1
n i – 2

**Moment Preserving Threshold** Color image processing by using binary quaternion-moment-preserving thresholding technique. This paper presents a new moment-preserving thresholding technique, called the binary quaternion-moment-preserving (BQMP) thresholding, for color image data. Based on representing color data by the quaternions, the statistical parameters of color data can be expressed through the definition of quaternion moments. Analytical formulas of the BQMP thresholding can thus be determined by using the algebra of the quaternions. The computation time for the BQMP thresholding is of order of the data size. By using the BQMP thresholding, quaternion-moment-based operators are designed for the application of color image processing, such as color image compression, multiclass clustering of color data, and sub pixel color edge detection. The experimental results show that the proposed operator for proposed edge operator can detect the color edge at the sub pixel level. Therefore, the proposed BQMP thresholding can be used as a tool for color image processing.

*5.3 Mathematical calculations*
Let p0 & p1 denote the fractions of the below & above threshold pixels in f respectively.
The first three moments of g are just:

Zi= gray values& preserving the first three moments: $\mu$

$$Mi = \sum pj(zj)I \qquad i = 1, 2, 3$$

Mi=mi I=1, 2, 3
p0+p1=1
After solving the above we get:

P0 is obtained & corresponding value of T is:

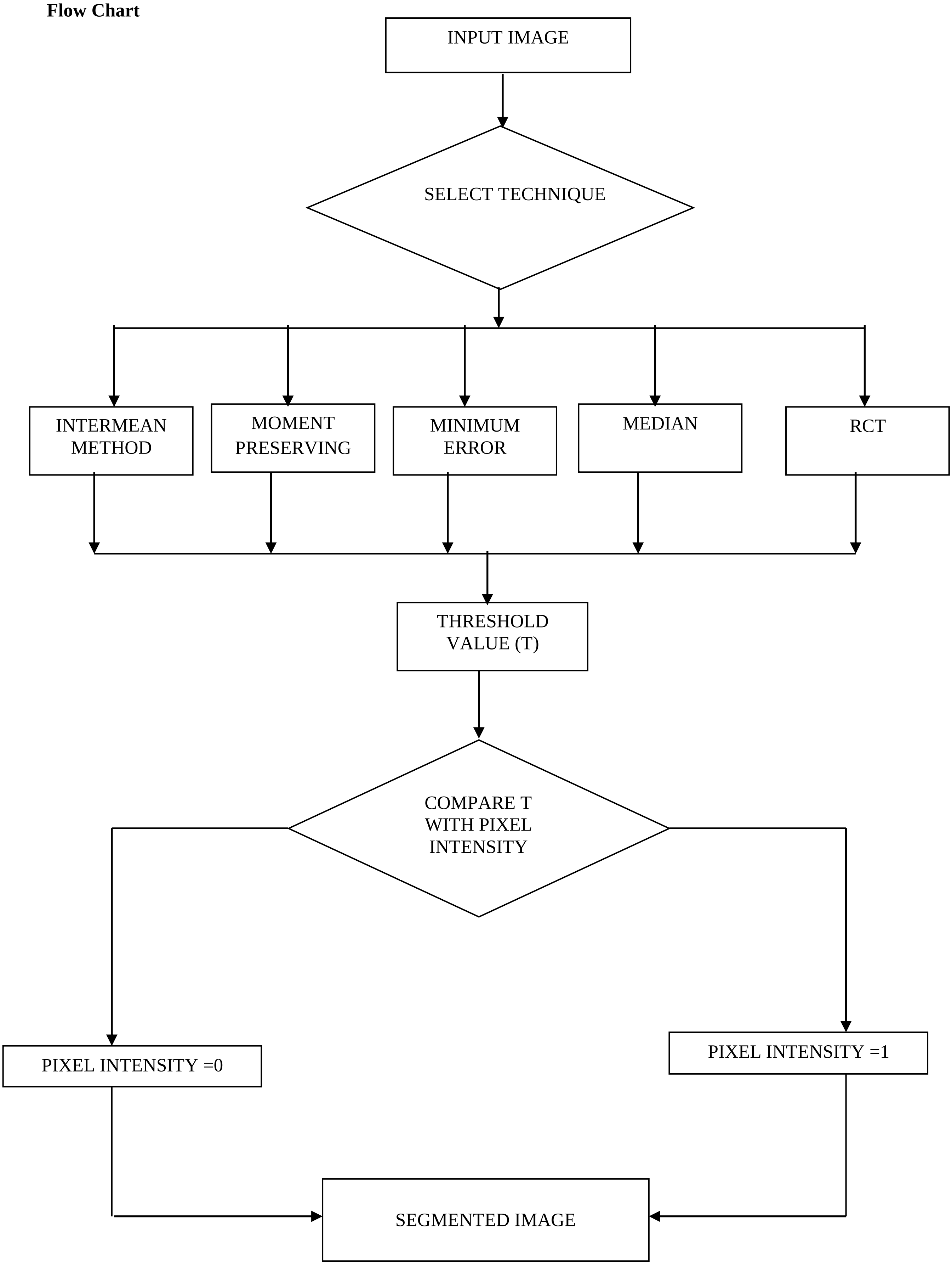
(1/n) p0 $\sum nj$

zj < 1

*5.4 Otsu Method:*
Otsu’s method is used in computer vision to perform thresholding.

**Flow Chart**


INPUT IMAGE
SELECT TECHNIQUE
INTERMEAN METHOD
MOMENT PRESERVING
MINIMUM ERROR
MEDIAN
RCT
THRESHOLD VALUE (T)
COMPARE T WITH PIXEL INTENSITY
PIXEL INTENSITY =0
PIXEL INTENSITY =1
SEGMENTED IMAGE

## VI. CONCLUSION

If each pixel p of an image has been assigned an 'interest' score, f(p) then Otsu creates a histogram off over the image; and selects a threshold to maximize the between-class variance. In statistics, a histogram is a graphical display of tabulated frequencies; A histogram is the graphical version of a table which shows what proportion of cases fall into each of several or many specified categories. The categories are usually specified as no overlapping intervals of some variable. The categories (bars) must be adjacent. The histogram is one of the seven basic tools of quality control, which include the histogram. Pareto chart, check sheet, control chart, cause and effect diagram, flowchart and scatter diagram.

## REFERENCES

[1] Masoud Alghoniemy and Ahmed H. Tewfik, "Geometric Distortion Correction through Image Normalization" IEEE Trans. Vol pp. 1291-1294, 2000

[2] M. Emre Celebi and Y. Alp Aslandogan, "A Comparative study of three moment based shape descriptors", Pattern recognition.http://www.emre.u-ta.edu/publications/ITCC2005_2.pdf

[3] D. S. Zhang and G. Lu. "Content-Based Shape Retrieval Using Different Shape Descriptors: A Comparative Study". In Proc. of IEEE International Conference on Multimedia and Expo (ICME2001), pp.317-320, Tokyo, Japan, August 22-25, 2001.

[4] Cho-Huak Teh and R.T Chin Member IEEE, "On image analysis by the methods of moments", IEEE Trans. Pattern Analysis and machine intelligence, vol. 10, pp 496-513, july 1998.

[5] Heloise Hse and A. Richard Newton, "Sketched Symbol Recognition Using Zernike Moments", Proceedings of the 17th international conference on pattern Recognition, ICPR 2004.

[6] Masoud Farzam and Shahram Shirani, "A Robust Multimedia Watermarking Technique Using Zernike Transform", IEEE, pp 529-534, 2001.

[7] Simon X.Liao and Miroslaw pawlak, "On the Accuracy of Zernike moments for image analysis", IEEE Trans. Pattern Analysis and machine intelligence, vol 20, pp 489-497, December 1998.

[8] Simon X.Liao and Miroslaw pawlak, "image analysis with Zernike moment descriptor", IEEE Trans. Pattern Analysis and machine intelligence, pp 700-703, 1997.

[9] S.Ghosal and R. Mehrotra, "Zernike moment-Based Feature Detectors", IEEE Trans., pp 934-938, 1994.

[10] Padilla, A. Martinez-Ramirez and F.Granados-Agustin, "Digital Image Reconstruction by using Zernike moments", Pattern recognition, 2003.http://www-optica.inaoep.mx/ investigadores /apadilla/memoria1.pdf

[11] Masoud Alghoniemy and Ahmed H. Tewfik, "Image Watermarking by Moment Invariants", IEEE Trans. Pattern Analysis and machine intelligence, pp 73-76, 2000.

[12] Ramakrishnan Mukundan, "A New Class Of Rotational Invariants Using Discrete Orthogonal Moments", Proceedings of the 6th IASTED International Conference, Signal And Image Processing, August 23-25, 2004.

[13] R. Gonzalez and R. E. Woods, Digital Image Processing, Prentice Hall, 2002.

[14] TIFF, Revision 6.0, (June 1992). Adobe Developers Association.www.adobe.com/ Support/TechNotes-.html

[15] http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/SHUTLER3/node1.html

[16] http://personal.gscit.monash.edu.au/~dengs/resource/papers/thesis_abstr act.html

[17] R. Boyle and R. Thomas *Computer Vision: A First Course*, Blackwell Scientific Publications, 1988, pp 35-41.

[18] R. Gonzalez and R. Woods *Digital Image Processing*, Addison-Wesley Publishing Company, 1992, Chap. 4.

[19] A. Jain *Fundamentals of Digital Image Processing*, Prentice-Hall, 1986, pp 241 - 243.

[20] A. Marion *an Introduction to Image Processing*, Chapman and Hall, 1991, Chap. 6.

[21] R. Boyle and R. Thomas *Computer Vision: A First Course*, Blackwell Scientific Publications, 1988, Chap 4.

[22] E. Davies *Machine Vision: Theory, Algorithms and Practicalities* Academic Press, 1990, Chap 4.

[23] A. Marion *an Introduction to Image Processing*, Chapman and Hall, 1991. Chap 5.

[24] D. Vernon *Machine Vision*, Prentice-Hall, 1991, p 49

[25] Image enhancement by histogram hyperrbolization. CGIP (6), No 3 June 1997, pp.286-294, BibRef7706.