Two Dimensional Clipping Based Segmentation Algorithm for Grayscale Fingerprint Images

Krishna Prasad K.[1] & P. S. Aithal[2]

[1]Research Scholar, College of Computer and Information Science, Srinivas University, Mangaluru-575001, Karnataka, India

[2] College of Computer and Information Science, Srinivas University, Mangaluru-575001, Karnataka, India

E-Mail: karanikrishna@gmail.com

Type of the Paper: Research Paper.
Type of Review: Peer Reviewed.
Indexed In: OpenAIRE.
DOI: http://dx.doi.org/10.5281/zenodo.1037627.
Google Scholar Citation: IJAEML

How to Cite this Paper:

Krishna Prasad, K. & Aithal, P. S. (2017). Two Dimensional Clipping Based Segmentation Algorithm for Grayscale Fingerprint Images. *International Journal of Applied Engineering and Management Letters (IJAEML)*, 1(2), 51-65.
DOI: http://dx.doi.org/10.5281/zenodo.1037627.

International Journal of Applied Engineering and Management Letters (IJAEML)
A Refereed International Journal of Srinivas University, India.

Disclaimer: The scholarly papers as reviewed and published by the Srinivas Publications (S.P.), India are the views and opinions of their respective authors and are not the views or opinions of the S.P. The S.P. disclaims of any harm or loss caused due to the published content to any party.

Two Dimensional Clipping Based Segmentation Algorithm for Grayscale Fingerprint Images

Krishna Prasad K.[1] & P. S. Aithal[2]

[1]Research Scholar, College of Computer and Information Science, Srinivas University, Mangaluru-575001, Karnataka, India

[2] College of Computer and Information Science, Srinivas University, Mangaluru-575001, Karnataka, India

E-Mail: karanikrishna@gmail.com

ABSTRACT

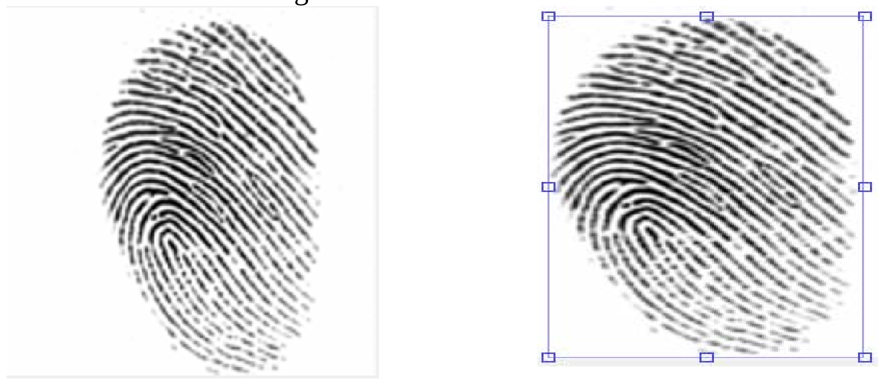
One of the huge methods in Automated Fingerprint Identification System (AFIS) is the segment or separation of the fingerprint. The process of decomposing an image into exclusive components is referred as segmentation. Fingerprint segmentation is the one of the predominant process involved in fingerprint pre-processing and it refers to the method of dividing or separating the image into disjoint areas as the foreground and the background region. The foreground also called as Region of Interest (ROI) due to the fact only the region which contains ridge and valley structure is used for processing, whilst the background carries noisy and irrelevant content material and so that it will be discarded in later enhancement or orientation or classification method. The challenge proper right here is to decide which a part of the image belongs to the foreground, retrieved as an input from the fingerprint sensor device or from benchmark datasets and which part belongs to the background. A 100% correct segmentation is continually very tough, specifically inside the very poor quality image or partial image together with the presence of latent. In this paper, we discuss a modified clipped based segmentation algorithm by adopting threshold value and canny edge detection techniques. We segment the background image is x and y dimensions or in other words left the edge, right edge, top edge and bottom edge of the image. For the purpose of analyzing the algorithm FVC ongoing 2002 benchmark dataset is considered. The entire algorithm is implemented using MATLAB 2015a. The algorithm is able to find affectively ROI of the fingerprint image or separates the foreground region from the background area of the fingerprint image very effectively. In high configuration system proposed algorithm achieves execution time of 1.75 seconds.

Keywords: Segmentation, Two Dimensional Clipping, Canny Edge Detection, Image Enhancement, Region of Interest (ROI).

1. INTRODUCTION :

Biometrics is an intrinsic bodily or behavioral characteristic that may be used to discover or verify the person. The most commonplace types of biometrics are face, speech, iris, fingerprint, gait, and signature. The fingerprint is very not unusual and popular biometric of type traits due to its universality, distinctiveness, and permanence and additionally, many advances and new researchers are to be had on this discipline. Despite the fact that AFIS is capable of recognizing a fingerprint image sample with already saved fingerprint image within the database, nevertheless, partial or latent fingerprint image suffers from the low-overall performance rate. A critical and crucial step with a purpose to obtain high first-class and overall performance at all sorts of the image is through correct segmentation. Fingerprints are generally labeled into three kinds as rolled, plain and latent fingerprints primarily based on the system, how they may be captured or accumulated [1]. In rolled fingerprint image is captured from one end of the finger to another end by rolling and mounting on capturing device in order to obtain complete ridge and valley details of the fingerprint. The plain fingerprint is directly captured using a fingerprint capturing device through pressing a fingertip onto a flat surface. Rolled and plain fingerprints are acquired in a sophisticated attended mode; they will be having good visual quality at the time of training and performance quality at the time of matching one to one or one to many for verification or identification purpose [2-5].

The emphasis is on ROI-segmentation is to accurately extract the ROI, which is real ridge details, which directly influences the performance of feature extraction and matching process. The example of ROI is shown below in Figure 1.



(a) Original Image

(b) ROI of the image (a)

Fig 1: Example of ROI

Usually, Latent fingerprints are collected from the crime scene and mixed with another image or components like structure noise or other fingerprints or on the surface of a wall that was inadvertently touched or handled. The algorithms work well for rolled and plain fingerprint shows significant flaws for latent image or suspect in identifying crime persons. Fingerprint segmentation is the one of the main process involved in fingerprint pre-processing and it refers to the process of dividing or separating the image into two disjoint regions as the foreground and background.

The simple method for segmentation of the fingerprint image is based on binarisation. Initially, the input image can be any of the type like rolled, plain or latent. In next step, the image has to resize using cropping the image or any other image resizing process. The fingerprint is usually in grayscale, but very rarely it can be color (RGB) image in the case of latent or any other types of partial or fingerprint captured using mobile devices. If the image is color, it should be converted into grayscale using RGB to Grayscale converter function. In next step, the Grayscale image is converted into a Binary image using coarse binarisation process. The purpose of coarse binarisation is to remove the background or noise associated with the input image or to separate foreground from the background image. Threshold method is used in order to get an initial binary image. To remove some background from the image, any threshold method can be used. Compared to the local adaptive threshold method, global methods are parameter independent and inexpensive [6-7]. Coarse scan supports detection of the fingerprint ridge positions from the background image. After the coarse binarisation process, the existence of false background is checked and if there exists (normal case), orientation angle is calculated using any orientation method, if not (abnormal case) again refined binarisation process is activated. The objective of refined binarisation is to find an optimal threshold to eliminate the background while preserving as much ridge pattern as possible. After all these processes, we get the segmented image.

In literature, a good number of papers are available for fingerprint segmentation, which can be roughly categorized under two classifications as block-wise methods and pixel-wise methods. In the block-wise method, the fingerprint images are classified into different equal sized nonoverlapping blocks and further organize blocks into a foreground region and background region based on the extracted features. On the other hand, pixel-wise methods emphasis on pixel and classifies the fingerprint image based on pixel-wise features of the image.

In this paper, we discuss a modified clipping based segmentation algorithm which clips white spaces or background image as left, right, top and bottom boundary. We discuss and analyze algorithm based on grayscale fingerprint image considered from FVC ongoing 2002 datasets.

2. LITERATURE REVIEW OF SEGMENTATION :

Segmentation or dividing is one of the deciders of performance in the automatic fingerprint recognition system. There is enough amount of literature with respect to image segmentation process or approach dating back over thirty years. Jain & Dubes, (1988) [8], explains the algorithm for clustering in his book, these early approaches for clustering can be used for segmentation, which acts as the basis for many new methods including boundary based segmentation such as Canny edge detection Canny, 1986 [9]. In this method, researcher defines a comprehensive set of goals for the computation of edge detection points. Adams and Bishof, (1994) [10], proposed segmentation algorithm for images, which are intensity images with certain characteristics like robust, rapid, and free of tuning parameters. This algorithm can take input as either individual pixels or regions and points these inputs to some region formed by the algorithm. The algorithms explain two methods in which input corresponds to the region, either by using manual seed or by an automated procedure. Chakraborty et al., (1996) [11], proposed a method which combines region based segmentation and boundary finding to form new method which is more robust to noise and high performance. The literature covered above is some general segmentation algorithms which will apply for any types of images.

In literature, there are many studies available, which mainly focuses on fingerprint image segmentation. Most of the segmentation algorithm does classification of the image based on either supervised learning or unsupervised learning. When a class label is not known or unknown, means of unsupervised learning, classification is significantly and very difficult. Researchers, Mehtre, et al. (1989) [12] classified the image into blocks, which is administrative specific and the size was 16 × 16 pixels. Based on the gradient distribution, each block was classified. This method is best suited for simple fingerprint images which contain only background and foreground. Later Researchers Mehtre and Chatterjee, (1989) [13] extended this work by leaving the grayscale variance, which will usually be lower than some threshold value. Researchers Ratha et al. (1995) [14] proposed 16 × 16 blocks of classes and each one was developed based on the gray scale variance in the direction opposite to the orientation of ridges.

The authors Jain and Ratha, (1997) [15] concentrated for the detection of objects located in complex backgrounds. The given object is first applied to a bank of even-symmetric Gabor filters. The output image received from the Gabor filter is subjected to a sigmoid function transformation. The yield image of the Gabor filter is applied as an input to the clustering algorithm, which develops spatially compact clusters. Sun and Ai (1996) [16] pre-processed initially fingerprint image by converting it into a binary image with the help of dynamic threshold value (T). Moayer and Fu (1975) [17] used sampling squares, which are obtained from the subdivision of fingerprint images for the ultimate goal of feature extraction. They used dynamic threshold value (T) to convert the initial image to a binary image. In order to determine the local threshold value, researchers used neighbor pixels by group 5 × 5 pixels.

Bazen and Gerez (2000) [18] used coherence and morphology of fingerprint image with an intention to obtain a smooth image by filtering different types of noises. The same author Bazen and Gerez (2001) [19] improved their work by adding two more statistical features as the mean and variance for their previous work. Here classification is done with the aid of optimal linear classifiers, which acts as a trainer for classification. With a goal to find compact cluster and reducing, classification error for post processing morphology is applied.

Naji et al. (2002) [20] developed a segmentation algorithm, which computerized or automated the method of selecting a threshold value at the time of segmentation with the aid of histogram equalizer. Segmentation algorithm generally falls under two categories of machine learning techniques as supervised learning and unsupervised learning. Unsupervised learning uses threshold decided on detecting features to cluster the image. Supervised learning uses a simple linear classifier to classify features as a region of interest (ROI) or background and foreground. As a part of supervised methods, Alonso-Fernandez et al. (2005) [21] used a Gabor filter to filter the input image and to obtain a smooth image. The neural network can also be used in the segmentation process to reduce the noise or to enhance the image quality.

Barreto et al. (2005) [22] used a neural network to train the fingerprint image data sets using Fourier spectrum and obtained a segmentation of fingerprint images. Zhu et al. (2006) [23] also used neural network concepts in order to train the fingerprint data set, but they used the gradient of the fingerprint orientation to segment the images. Wu et al. (2007) [24] proposed a new method for segmentation; in their method, they used the strength of Harries corner function to extract the background from foreground or to extract a region of interest. In order to separate region of interest from the background image, they used corner strength measures. Tiwari, K., & Gupta, P. (2015) [25] proposed a new method for extracting a single fingerprint image from the slap fingerprint scanner, which simultaneously scans four fingerprints of a person in a single image. While extracting the single fingerprint image the image is also required to be segmented. They used a novel technique to extract solitary (single) fingerprint image based on force field and heuristics using divide and conquer strategy and is tested in IITK-4slap-Rural and IITK-4slap-student database.

Thai, Huckemann, & Gottschlich, 2016 [26] proposed the new approach for fingerprint segmentation in three folds, firstly used factorized directional bandpass (FDB) and directional Hilbert transform originated from Butterworth bandpass (DHBB) filter combined with soft-thresholding for texture extraction. Secondly, as an evaluation benchmark with 10560 images marked manually for ground truth segmentation. Thirdly they have compared systematically factored directional filtering with other similar fingerprint segmentation approach and obtained comparatively good performance.

3. TWO DIMENSIONAL CLIPPING BASED SEGMENTATION-DESCRIPTION :

This algorithm considers binary fingerprint image, which is referred as I_{binary} . Initially to find the edges of the I_{binary} image efficiently canny edge detection method is used. Canny edge detection finds the edges of the image through different processes, which includes, smoothing, locating gradients, non-maximum suppression, double thresholding, and edge tracking by using hysteresis. Smoothing of the image is done with the help of convolution, which blurs the image to get rid of the noise. Canny edge detection uses double thresholding in order to find edges of the image. The result of the canny edge detection method is stored as I_{canny} . Next, the edge detected image, I_{canny} is converted into the-low resolution image by converting 256 × 256 sized grayscale images to 128 × 128 sized grayscale image.

$I_{LRE} = I_{binary}\ (i \times 2, j \times 2)$

The low-resolution image is represented as I_{LRE}. In the next phase I_{LRE} image is padded with zeros using pad array and usually for simplicity in this method we use pad array size is eight and is referred as P. For $I_{LRE,}$ eight zeros are added to row and column respectively, and it enhanced to 144 × 144 sized grayscale images, which is denoted as I_{Parray} .

The, I_{Parray} is clipped into 15×15 sized image and processed. The clipped image is stored in temp1. The entire 225 pixels of temp1 are reshaped as 1×225 matrix and denoted as temp2. The covariance of the matrix of the image, temp2 is calculated and if it is less than the threshold then the pixel of the I_{binary} (256 × 256 sizes) image is considered as not a part of ROI or foreground. Covariance of a matrix is calculated by considering row as observations and columns as random variables. Every pixel of the I_{binary} image is traced like this and marked as either foreground or background of the image based on covariance value. If it is greater than the threshold value then the pixel is considered to be foreground, means which is a real part of the fingerprint image. Each time when padarray is considered, this takes into account one pixel out of 128×128 low-resolution image and two pixels out of 256 ×128 sized image.

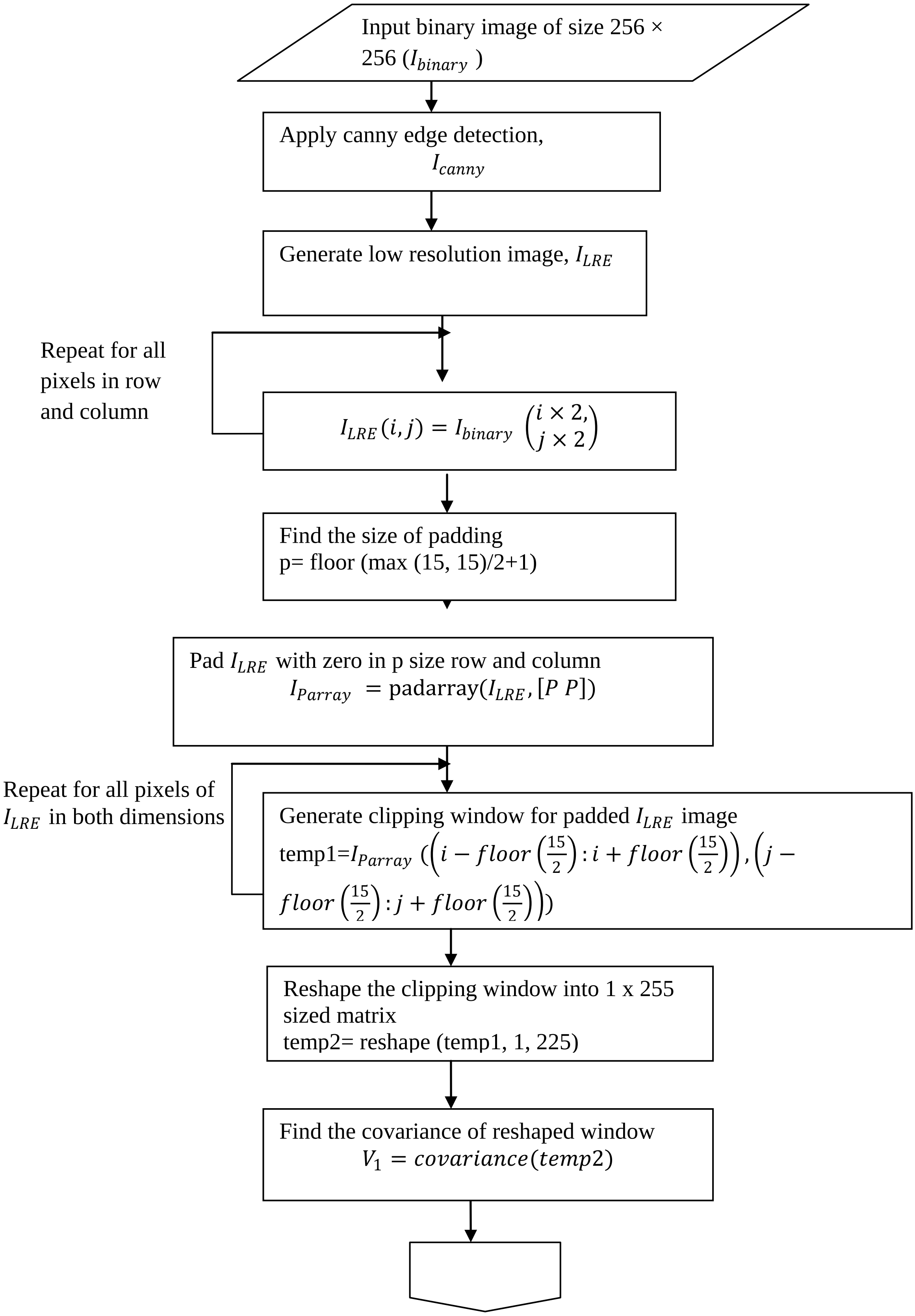
As the algorithm name suggests two-dimensional clipping, we discard maximum background part of the image by checking whether all the pixels of the each column intensity value sum become 256. If the column sum is 256 means all the pixels of that particular column contains intensity value 1. This signifies that this column contains background of the image. If any one column intensity value sum leads to value less than 256, which signifies that the particular column contains part of the foreground or ROI of the image. Then we skip the iteration and count considering starting of the column pixel for output of the segmented image from just previous to that column number. The same process we repeat from the last column to the first column in reverse direction and stop moving backward until we get a column number sum of intensity value less than 256 for the purpose of finding last column number, which contains at least one pixel of foreground pixel. This means that from the last column to till this position image contains only background part of the image. The above-mentioned method repeated for rows also. So that it eliminates background or white blank area in left edge, right edge, top edge and bottom edge regions.

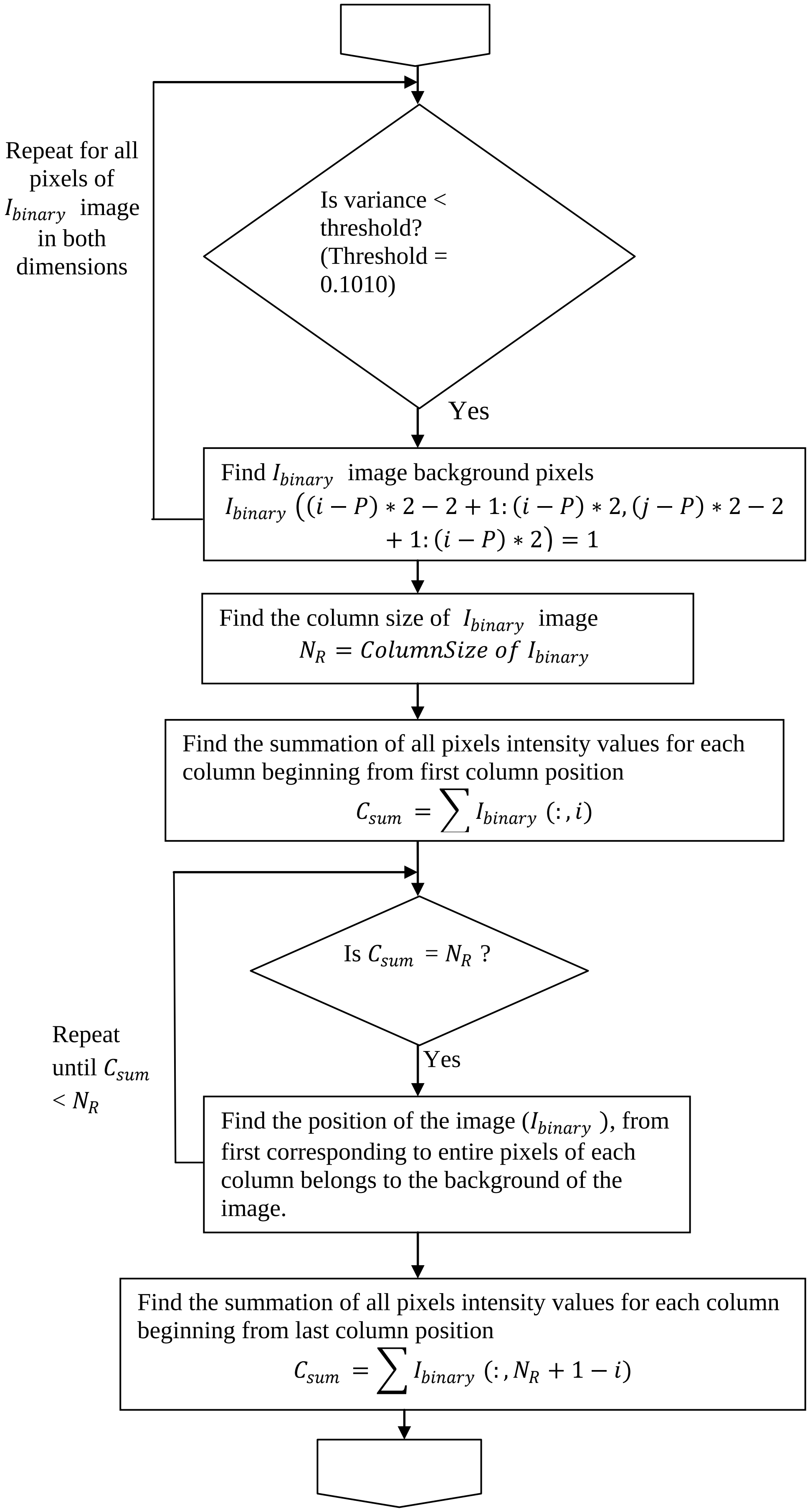
4. TWO DIMENSIONAL CLIPPING BASED SEGMENTATION-ALGORITHM :

```
Input: binary image, $I_{binary}$
Output: Segmented Image, $I_{segment}$
Step-1: Read $I_{bin ary}$ image
Step-2: Apply canny edge formation to the $I_{binary}$ and store it in a variable $I_{canny}$
Step-3: for i=1 to floor(R/2)
Step-4: for j=1 to floor (C/2)
Step-5: $I_{LRE} = I_{binary}\ (i \times 2, j \times 2)$; end for loop; $I_{LRE}$ → Low Resolution Edge Image
Step-6: $[R_{LRE}, C_{LRE}] = size(I_{LRE})$
Step-7: P=floor (max (15, 15)/2+1);
Step-8:$I_{Parray}$ = padarray($I_{LRE}$, [P P])
Step-9: for i=P+1 to $R_{LRE}$+P
Step-10: for j=P+1 to $C_{LRE}$+P
Step-11: temp1=$I_{Parray}$ ($\left(i - floor\left(\frac{15}{2}\right) : i + floor\left(\frac{15}{2}\right)\right), \left(j - floor\left(\frac{15}{2}\right) : j + floor\left(\frac{15}{2}\right)\right)$)
Step-12: temp2=reshape (temp1, 1, 225)
Step-13: $V_1 = covariance(temp2)$
Step-14: if $V_1 < T$                    \\ T → Threshold value=0.101
Step-15: $I_{binary}\ \left((i - P) * 2 - 2 + 1 : (i - P) * 2, (j - P) * 2 - 2 + 1 : (i - P) * 2\right) = 1$
Step-16: end if; end for;
Step-17: $N_R = ColumnSize\ of\ I_{binary}$
Step-18: for i=1 to $N_R$
Step-19: $C_{sum} = \sum I_{binary}\ (:, i)$
Step-20: Check if $C_{sum} = N_R$
Step-21: Position$_1 = i$; end if
Step-22: for i=1 to $N_R$
Step-23: $C_{sum} = \sum I_{binary}\ (:, N_R + 1 - i)$
Step-24: Check if $C_{sum} = N_R$
Step-25: Position$_2 = \frac{(N_R{}^2 - i^2)}{N_R + i}$; end if; end step-22 for
Step-26: $N_C = RowSize\ of\ I_{binary}$
Step-27: for i=1 to $N_R$
Step-28: $R_{sum} = \sum I_{binary}\ (:, i)$
Step-29: Check if $R_{sum} = N_C$
Step-30: Position$_3 = i$; end if
Step-31: for i=1 to $N_C$
Step-32: $R_{sum} = \sum I_{binary}\ (N_C + 1 - i, :)$
Step-33: Check if $R_{sum} = N_C$
Step-34: Position$_4 = \frac{(N_C{}^2 - i^2)}{N_C + i}$; end if; end step-27 for
```

Step-35:$I_{segment} \; = I_{binary}$ (Position$_3$ to Position$_4$: , Position$_1$ to Position$_2$)

5. FLOWCHART OF TWO DIMENSIONAL CLIPPING BASED SEGMENTATION :


Input binary image of size 256 × 256 (I_{binary})
Apply canny edge detection, I_{canny}
Generate low resolution image, I_{LRE}
Repeat for all pixels in row and column
$I_{LRE}(i,j) = I_{binary}\begin{pmatrix} i \times 2, \\ j \times 2 \end{pmatrix}$
Find the size of padding
p= floor (max (15, 15)/2+1)
Pad I_{LRE} with zero in p size row and column
$I_{Parray} \; = \text{padarray}(I_{LRE}, [P\ P])$
Repeat for all pixels of I_{LRE} in both dimensions
Generate clipping window for padded I_{LRE} image
temp1=I_{Parray} ($\left(\left(i - floor\left(\frac{15}{2}\right) : i + floor\left(\frac{15}{2}\right)\right), \left(j - floor\left(\frac{15}{2}\right) : j + floor\left(\frac{15}{2}\right)\right)\right)$)
Reshape the clipping window into 1 x 255 sized matrix
temp2= reshape (temp1, 1, 225)
Find the covariance of reshaped window
$V_1 \; = covariance(temp2)$



Repeat for all pixels of I_{binary} image in both dimensions
Yes
Yes
Repeat until C_{sum} $< N_R$

Is variance < threshold? (Threshold = 0.1010)

Find I_{binary} image background pixels

$$I_{binary}\left((i-P)*2-2+1:(i-P)*2,(j-P)*2-2+1:(i-P)*2\right)=1$$

Find the column size of I_{binary} image

$$N_R = ColumnSize\ of\ I_{binary}$$

Find the summation of all pixels intensity values for each column beginning from first column position

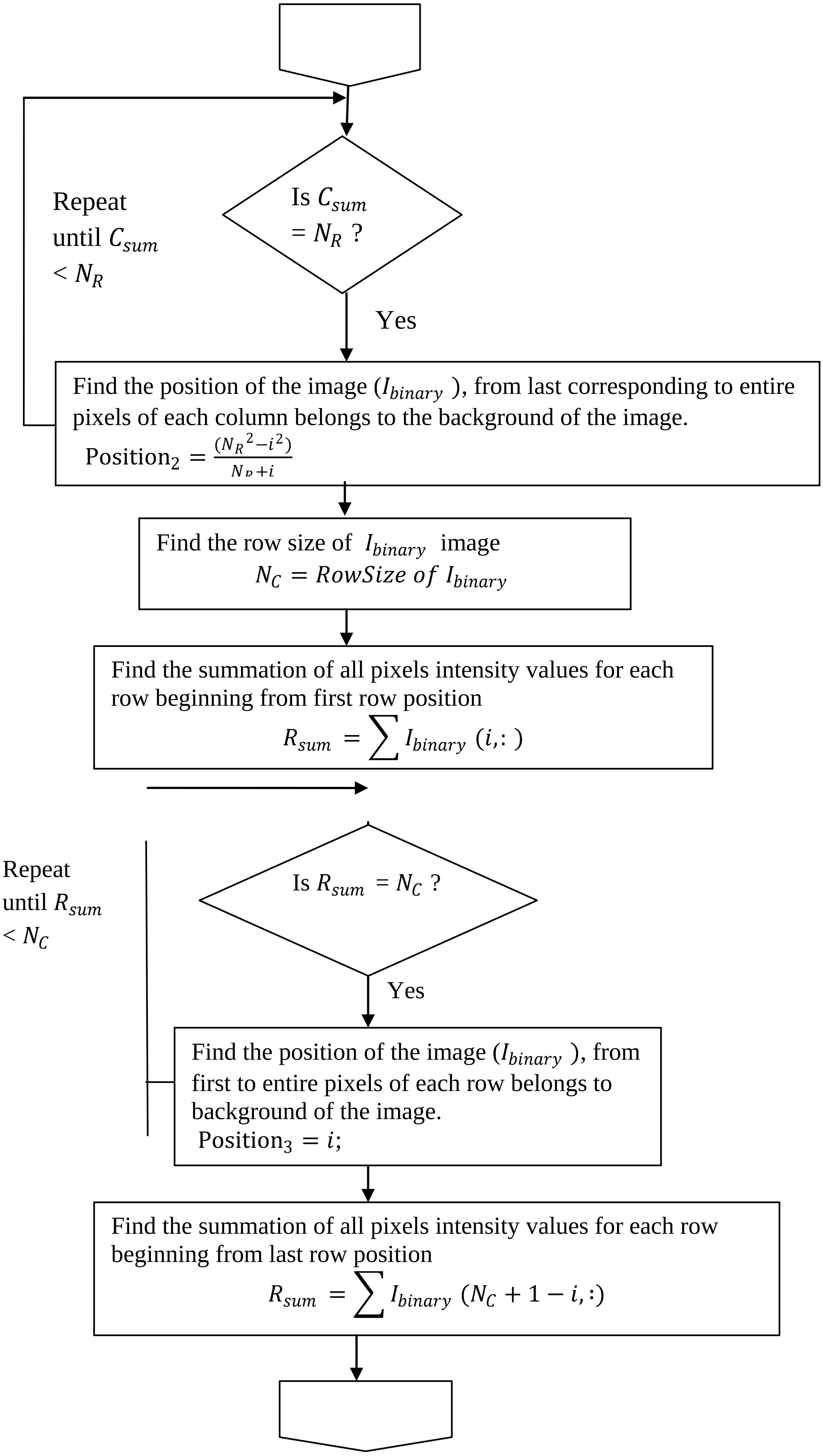
$$C_{sum} = \sum I_{binary}\,(:,i)$$

Is $C_{sum} = N_R$?

Find the position of the image (I_{binary}), from first corresponding to entire pixels of each column belongs to the background of the image.

Find the summation of all pixels intensity values for each column beginning from last column position

$$C_{sum} = \sum I_{binary}\,(:,N_R+1-i)$$


Repeat until $C_{sum} < N_R$
Is $C_{sum} = N_R$?
Yes
Find the position of the image (I_{binary}), from last corresponding to entire pixels of each column belongs to the background of the image.
$\text{Position}_2 = \frac{(N_R{}^2 - i^2)}{N_R + i}$
Find the row size of I_{binary} image
$N_C = RowSize\ of\ I_{binary}$
Find the summation of all pixels intensity values for each row beginning from first row position
$R_{sum} = \sum I_{binary}\ (i,:)$
Repeat until $R_{sum} < N_C$
Is $R_{sum} = N_C$?
Yes
Find the position of the image (I_{binary}), from first to entire pixels of each row belongs to background of the image.
$\text{Position}_3 = i;$
Find the summation of all pixels intensity values for each row beginning from last row position
$R_{sum} = \sum I_{binary}\ (N_C + 1 - i,:)$

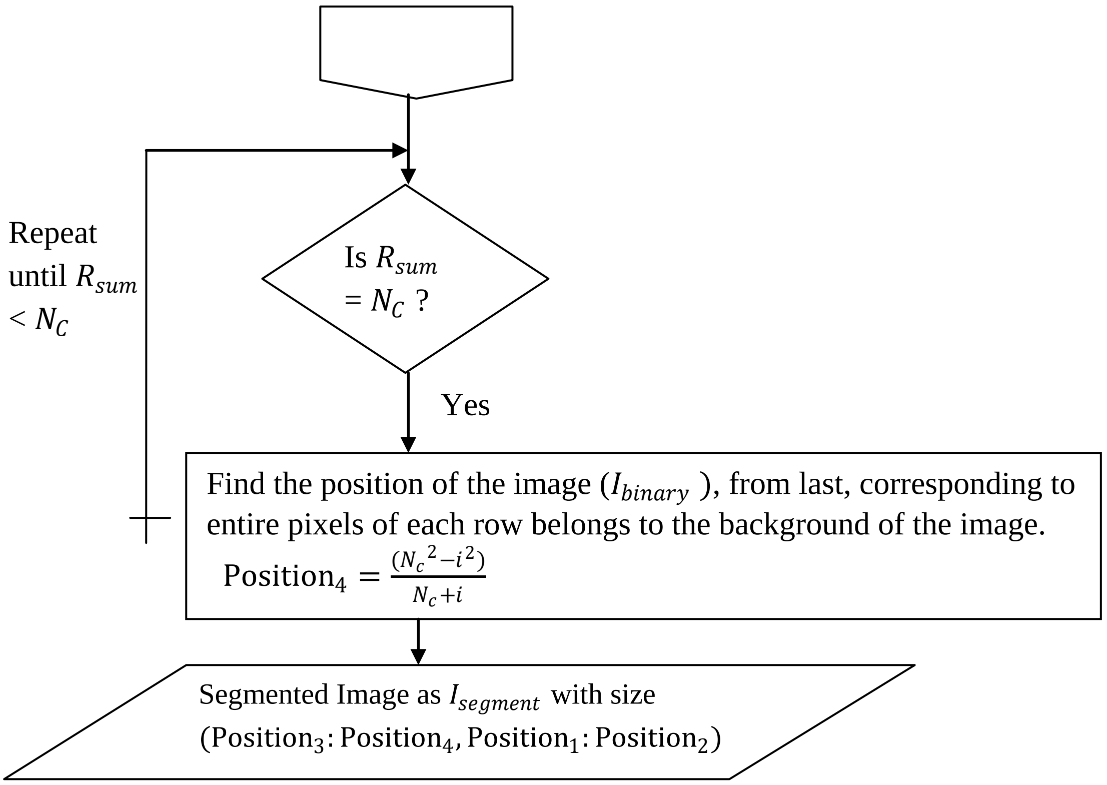

Repeat until $R_{sum} < N_C$
Is $R_{sum} = N_C$?
Yes
Find the position of the image (I_{binary}), from last, corresponding to entire pixels of each row belongs to the background of the image.
$\text{Position}_4 = \frac{(N_c{}^2 - i^2)}{N_c + i}$
Segmented Image as $I_{segment}$ with size
$(\text{Position}_3 : \text{Position}_4, \text{Position}_1 : \text{Position}_2)$

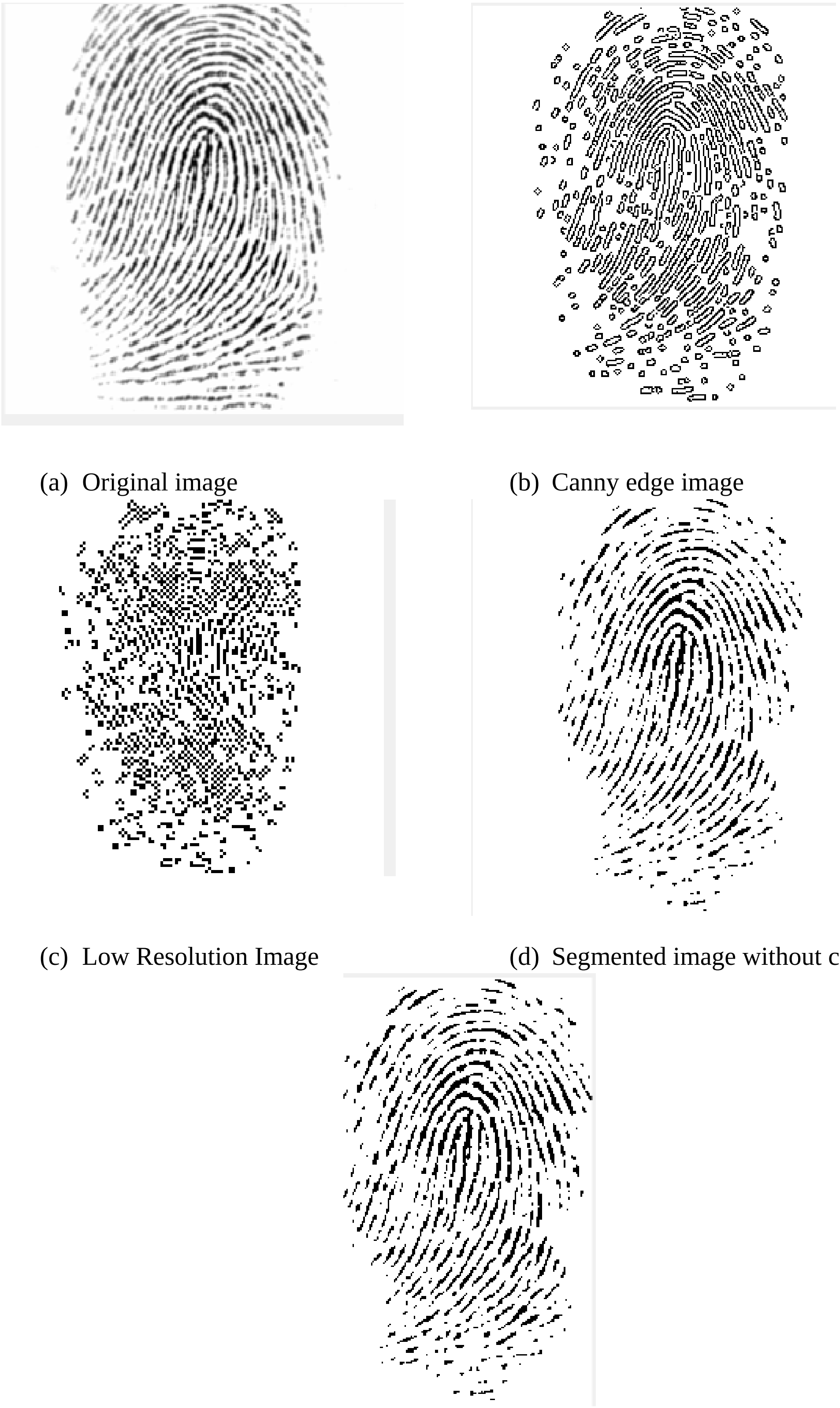
Fig. 2: Flowchart of Two dimensional clipping based segmentation algorithm

This proposed algorithm for segmentation is explained using a flowchart. The input for this algorithm is a binary image of size 256 × 256 which is represented as I_{binary} . The final output is segmented image denoted as, $I_{segment}$. The different workflows of the proposed algorithm are listed out below.

- Find the edge of the image using canny edge detection method
- Generate the low-resolution image from the 256 × 256 sized grayscale image
- Find the size of the padding.
- Pad the low-resolution image with zero along row and column directions.
- Generate the clipping window of size 15×15 for the low-resolution padded image.
- Reshape the clipping widow as 1 × 256 size window
- Find covariance of the matrix of the image for each element of the input image through the padded image.
- If covariance is less than the threshold treat it as background image otherwise treat it as the foreground image.
- Discard the each column of the image if it completely contains the intensity value 1 from both left and right directions of the image.
- Discard the each row of the image if it completely contains the intensity value 1 from both top and bottom directions of the image.

6. ANALYSIS OF THE TWO DIMENSIONAL CLIPPING BASED SEGMENTATION :

The two dimensional clipping based Segmentation is analysed by considering FVC ongoing 2002 DB1_B datasets. A sample fingerprint image named as 102_1.tif from FVC ongoing 2002 dataset is considered in figure 3.15. The algorithm is implemented using MATLAB programming version 2015a.



(a) Original image

(b) Canny edge image

(c) Low Resolution Image

(d) Segmented image without clipping

(e) Segmented image by clipping left, right, top and bottom border

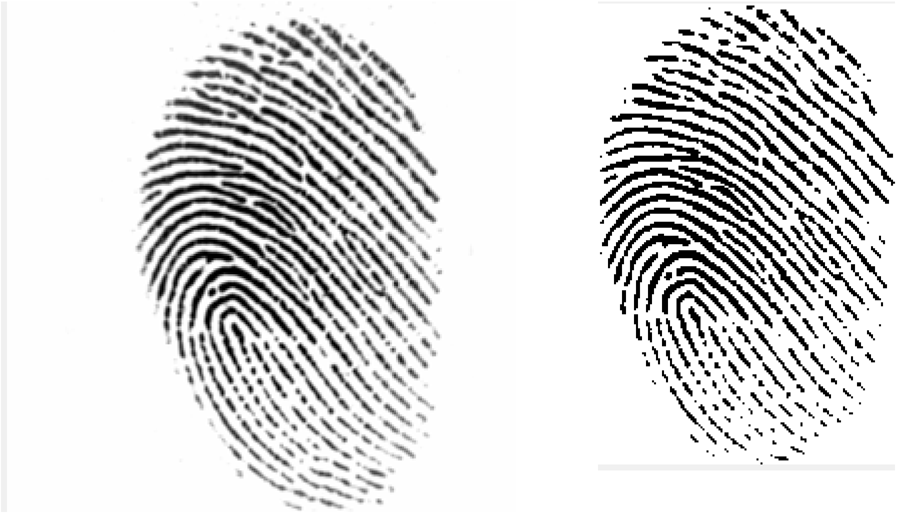
Fig. 3: Examples of Two dimensional Clipping based segmentation algorithm different phase's results

Table 1, shows edges obtained using canny edge detection method. Total numbers of edges are considered in terms of a total number of pixels.

Table 1: Total number of edges identified through canny edge detection

Sr. No	**Image name**	**Total number of Edges identified using canny edge Detection**
1	101_1.tif	8774
2	102_1.tif	8302
3	103_5.tif	10023
4	104_4.tif	10402
5	105_8.tif	7354
6	101_6.tif	10921
7	103_2.tif	10136

Figure 4 shows input image, and results of segmentation process using two dimensional clipping based segmentation for different sample images of FVC ongoing DB1_B datasets. While seeing the two images we don't find any differences. But if we observe carefully after segmentation left and right part of the filtered image is clipped, which corresponds to background pixels.



(a) Input image

(b) Segmented image

Fig. 4: Examples of input and segmented image using proposed methods

Table 2 shows the total number of pixels before the segmentation and after segmentation. If the numbers of pixels are reduced, this improves the execution performance or speed in following stages of automatic fingerprint identification systems like minutiae formation, feature extraction, and matching.

Table 2: Comparison of total number of pixels before and after segmentation

Sr. No	**Image name**	**Size of the image before segmentation**	**Size of the image after segmentation**	**Total number of pixels before segmentation**	**Total number of pixels after segmentation**
1	101_1.tif	256 × 256	230 × 148	65536	34040
2	102_1.tif	256 × 256	251 × 149	65536	37399
3	103_5.tif	256 × 256	228 × 183	65536	41724
4	104_4.tif	256 × 256	222× 193	65536	42846
5	105_8.tif	256 × 256	236 × 153	65536	36108
6	101_6.tif	256 × 256	213 × 180	65536	38340
7	103_2.tif	256 × 256	256 × 150	65536	38400

Table 3 shows the execution time of two dimensional clipping based segmentation algorithm.

Execution time is calculated based on duration or time between input and output. Execution time is calculated on two different configuration laptops. One is referred as System-I and other is System-II. The configuration of System-I and System-II are given in Table 4.

Table 3: Execution time for different sample images using two dimensional clipping based segmentation

Sr. No	Image name	Execution Time in Seconds Using System-I	Mean Execution time using System-I	Execution Time in Seconds Using System-II	Mean Execution time using System-II
1	101_1.tif	4.019798	3.960607	1.741521	1.754737
2	102_1.tif	4.011555		1.815977	
3	103_5.tif	3.931000		1.712644	
4	104_4.tif	3.946413		1.669623	
5	105_8.tif	3.933841		1.742137	
6	101_6.tif	3.866910		1.676781	
7	103_2.tif	4.014733		1.924482	

Table 4: Configuration of System-I and Sytem-2 used for finding Execution Time

Sr. No.	Parameters	System-I	System-II
1	Model	Compaq 435	TravelMate 5742
2	Processor	AMD E-350 processor 1.60 GHz	Intel (R) Core (TM) i3 CPU, M 370 @ 2.40 GHz
3	Installed Memory	3 GB (2 GB usable)	6 GB (5.68 GB usable)
4	System Type	32-bit operating System	64-bit operating System
5	Operating System	Windows 7 Starter	Windows 7 Professional
6	Software	MATLAB 2015a 32-bit	MATLAB 2015a 32-bit

Execution time for the two dimensional clipping based segmentation algorithm in System-I is almost near to 4 seconds and in System-II is 1.75 seconds. So it can be proved that execution time is depending on the configuration of the system. If the system is high configured system in terms of processor, memory etc. The algorithm executes faster than the system which is having less configuration.

7. CONCLUSION :

An essential and important step in order to obtain high quality and performance rate at all types of image is through accurate segmentation. Fingerprint segmentation is the one of the main process involved in fingerprint pre-processing and it refers to the process of dividing or separating the image into two disjoint regions as the foreground and background.

The Two dimensional clipping based segmentation algorithm affectively clips the background region of the fingerprint in all four permissible boundaries, left edge, right edge, top edge and bottom edge. The use of canny edge detection method improves the identification of the edges effectively. The proposed algorithm has good execution time in high configured systems. The proposed algorithm has following characteristics.

- Usage of canny edge detection techniques finds all edges of the image efficiently.
- Having the ability to generate the low resolution image from the 256 × 256 sized grayscale image
- Pads the low resolution image with zero along row and column directions.
- Generate the clipping window of size 15×15 for low resolution padded image.
- Reshape the clipping widow as 1 × 256 size window
- Find covariance of the matrix of the image for each element of the input image through the padded image.

- If covariance is less than the threshold treats it as background image otherwise treat it as the foreground image.
- Discards each column of the image, if it completely, contains the intensity value 1 from both left and right directions of the image.
- Discards each row of the image, if it completely, contains the intensity value 1 from both top and bottom directions of the image.

REFERENCES :

[1] Zhang, J., Lai, R., & Kuo, C. C. J. (2012). Latent fingerprint detection and segmentation with a directional total variation model. In Proceedings - International Conference on Image Processing, ICIP, 1145–1148. DOI: https://doi.org/10.1109/ICIP.2012.6467067

[2] Krishna Prasad, K. & Aithal, P. S. (2017). A Conceptual Study on Image Enhancement Techniques for Fingerprint Images. *International Journal of Applied Engineering and Management Letters (IJAEML),* 1(1), 63-72. DOI: http://dx.doi.org/10.5281/zenodo.831678

[3] Krishna Prasad, K. & Aithal, P. S. (2017). Literature Review on Fingerprint Level 1 and Level 2 Features Enhancement to Improve Quality of Image. *International Journal of Management, Technology, and Social Sciences (IJMTS),* 2(2), 8-19. DOI: http://dx.doi.org/10.5281/zenodo.835608

[4] Krishna Prasad, K. & Aithal, P. S. (2017). Fingerprint Image Segmentation: A Review of State of the Art Techniques. *International Journal of Management, Technology, and Social Sciences (IJMTS),* 2(2), 28-39. DOI: http://dx.doi.org/10.5281/zenodo.848191

[5] Krishna Prasad, K. & Aithal, P. S. (2017). A Novel Method to Contrast Dominating Gray Levels during Image contrast Adjustment using Modified Histogram Equalization. *International Journal of Applied Engineering and Management Letters (IJAEML),* 1(2), 27-39. DOI: http://dx.doi.org/10.5281/zenodo.896653

[6] Wang, Q. P., Du, J. X., & Zhai, C. M. (2010). Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* 6216(February 2016), 240–246. DOI: https://doi.org/10.1007/978-3-642-14932-0

[7] Vielhauer, C., Dittmann, J., Drygajlo, A., Juul, N. C., & Fairhurst, M. (Eds.). (2011). Biometrics and ID Management: COST 2101 European Workshop, BioID 2011, Brandenburg (Havel), March 8-10, 2011, Proceedings (Vol. 6583). Springer Science & Business Media.

[8] Jain, A. K., & Dubes, R. C. (1988). Algorithms for clustering data. Prentice-Hall, Inc.

[9] Canny, J. (1986). A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence, (6), 679-698.

[10] Adams, R., & Bischof, L. (1994). Seeded region growing. IEEE Transactions on pattern analysis and machine intelligence, 16(6), 641-647.

[11] Chakraborty, A., Staib, L. H., & Duncan, J. S. (1996). Deformable boundary finding in medical images by integrating gradient and region information. IEEE Transactions on Medical Imaging, 15(6), 859-870.

[12] Mehtre, B. M., Murthy, N. N., Kapoor, S., & Chatterjee, B. (1987). Segmentation of fingerprint images using the directional image. Pattern Recognition, 20(4), 429-435.

[13] Mehtre, B. M., & Chatterjee, B. (1989). Segmentation of fingerprint images—a composite method. Pattern Recognition, 22(4), 381-385.

[14] Ratha, N. K., Chen, S., & Jain, A. K. (1995). Adaptive flow orientation-based feature extraction in fingerprint images. Pattern Recognition, 28(11), 1657-1672.

[15] Jain, A. K., Ratha, N. K., & Lakshmanan, S. (1997). Object detection using Gabor filters. Pattern Recognition, 30(2), 295-309.

[16] Sun, X. and Ai, Z. (1996) Automatic feature extraction and recognition of fingerprint images, Proceeding of ICSP'96, Beijing, Pp.1086-1089.

[17] Moayer, B., & Fu, K. S. (1975). A syntactic approach to fingerprint pattern recognition. Pattern Recognition, 7(1–2), 1–23. DOI: https://doi.org/10.1016/0031-3203(75)90011-4

[18] Bazen, A.M. and Gerez, S.H. (2000) Directional field computation for fingerprints based on the principal component analysis of local gradients, Proceedings of ProRISC2000, 11th Annual Workshop on Circuits, Systems and Signal Processing, Veldhoven, The Netherlands.

[19] Asker M. Bazen and Sabih H. Gerez. 2001. Segmentation of Fingerprint Images, Workshop on Circuits, Systems and Signal Processing, Veldhoven. The Netherlands.

[20] Naji, A.W., Ramli, A.R., Ali, R., Rahman, S.A., and Ali, M.L. (2002) A segmentation algorithm based on histogram equalizer for fingerprint classification system, Second International Conference on Electrical and Computer Engineering ICECE 2002, Dhaka, Bangladesh, pp. 390-393.

[21] Alonso-Fernandez, F., Fierrez-Aguilar, J. and Ortega-Garcia, J. (2005) An enhanced Gabor filter based segmentation algorithm for fingerprint recognition systems, In Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, Pp. 239-244.

[22] Barreto, P., Marques, A.C. and Thome, A.C. (2005) A neural network fingerprint segmentation method, 5th International Conference on Hybrid Intelligent Systems P.6.

[23] Zhu, E., Yin, J., Hu, C. and Zhang, G. (2006) A systematic method for fingerprint ridge orientation estimation and image segmentation, Pattern Recognition, Vol. 39, No.8, Pp. 1452-1472.

[24] Wu C., Tulyakov S. and Govindaraju V. (2007). Robust point-based Feature Fingerprint Segmentation Algorithm, ICB (2007), Pp. 1095-1104

[25] Tiwari, K., & Gupta, P. (2015). An efficient technique for automatic segmentation of fingerprint ROI from digital slap image. Neurocomputing, 151(P3), 1163–1170. https://doi.org/10.1016/j.neucom.2014.04.086

[26] Thai, D. H., Huckemann, S., & Gottschlich, C. (2016). Filter design and performance evaluation for fingerprint image segmentation. PLoS ONE, 11(5). DOI: https://doi.org/10.1371/journal.pone.0154160

****