TOPOLOGY PRESERVING SKELETONIZATION TECHNIQUES FOR GRAYSCALE IMAGES

[1]Dr. A. Sri Krishna, [2]Dr. K. Gangadhar, [3]N. Neelima, [4]K. Ratna Sahithi

[1,3,4]Dept. of IT, R.V.R& J.C College of Eng. Guntur, India.
[2]Acharya Nagarjuna University

Abstract— Thinning is an interesting and challenging problem, and plays a central role in reducing the amount of information to be processed during pattern recognition, image analysis and visualization, computer-aided diagnosis. A topology preserving thinning algorithm is used which removes the pixels from a gray scale images. First it checks for the condition whether a pixel is acyclic or not. If a pixel is acyclic, then it is removed from the image else retains in the image. A topology preserving skeleton is a synthetic representation of an object that retains its shape, topology, geometry, connectivity and many of its significant morphological properties. It has been observed in our experiments that thinning algorithm is stable and robust and yield promising performance for wide range of images.

Keywords— ***Skeletonization, Topology preservation, Thinning.***

I. INTRODUCTION

Thinning is a widely used pre-processing step in digital image processing and pattern recognition. It is iterative layer by layer peeling, until only the “skeletons” of the objects are left. Thinning algorithms are generally constructed in the following way: first the thinning strategy and the deletion rules are figured out, then the topological correctness is proved[3]. In the case of the proposed algorithms, first consider some sufficient conditions for parallel reduction operators to preserve topology[26], and then the deletion rules were accommodated to them.

Thinning is the process of reducing an object in a digital image to the minimum size necessary for machine recognition of that object[28]. After thinning, analysis on the reduced size image can be performed as shown in Fig.1. Thinning is essentially a “pre-processing” step used in many image analysis techniques. The thinning process reduces the width of pattern to just a single pixel. Thinning when applied to a binary image, produces another binary image[1].

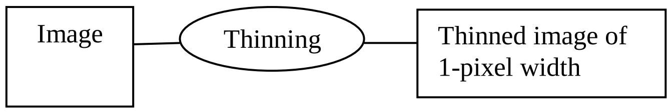


Fig. 1. Thinning block diagram

Thinning has been used in a wide variety of applications including: pattern recognition, medical imaging analysis, bubble-chamber image analysis (a device for viewing microscopic particles), text and handwriting recognition and analysis[6]. A thinning algorithm is said to be good if it posses the following set of desirable features: maintaining connectivity of resulting skeletons; producing skeletons of unit width; insensitive to noise; time efficient and preserving topology[26].

The paper is organized as follows. In Section II, prior work is discussed. Topology Preserving Skeletonization algorithm is presented in Section III. Section IV consists of results and discussions. Conclusion is given in Section V.

II. PRIOR WORK

Skeletonization is a process for reducing foreground regions in a binary image to a skeletal remnant that largely preserves the extent and connectivity of the original region while throwing away most of the original foreground pixels. It (i.e., skeleton extraction from a digital binary picture) provides region-based shape features[21]. It is a common preprocessing operation in pattern recognition or in raster-to-vector conversion.

The three skeletonization techniques are :

- Ridges are detected in distance map of the boundary points.

- Calculating the Voronoi diagram generated by the boundary points, and
- Thinning is nothing but layer by layer peeling.

In digital spaces, only an approximation to the "true skeleton" can be extracted. The two requirements that are to be considered is shown below:

- Topological (to maintain the topology of the original object),
- Geometrical (forcing the "skeleton" being in the middle of the object and invariance under the most important geometrical transformation including translation, rotation, and scaling).

Distance transformation:

Skeletonization using distance transformation consists of three steps:

- The original (binary) image is converted into feature and non-feature elements. The boundary of the object contains feature elements.
- Generating the distance map where each element gives the distance to the nearest feature element.
- The ridges are detected as skeletal points.

The idea behind our ridge detection algorithm is based on the well-known fact that the gradient at any point on a distance map generally points towards the ridge and reverses its direction as it crosses the ridge. In other words, for a point to be on a ridge, it must be a local maximum on some direction, i.e., on a line passing through the point bearing that direction. Consider a line with arbitrary orientation passing through a point of a distance map. If the point is a local maximum in the direction of the line, the distance values of the point's two opposite neighboring points must be less than that of the point, and the directions of the two opposite neighbors' gradient vectors projected onto the line must be opposite, pointing to the given point. In short, the given point generates a sign barrier between the two opposite neighbors on the line, if it is on a ridge of a distance map. To determine the minimum number of orientations, we need to understand when a particular orientation of the line succeeds or fails to detect certain ridges.

There are two ways that a ridge interacts with a projection line: a ridge either intersects or does not intersect the line. When a ridge intersects the projection line, it generates a sign barrier between the two points on the line enclosing the ridge. In other words, a ridge is guaranteed to be detected by the sign barrier on the projection line if it intersects the line. In contrast, if a ridge is nearly parallel to the projection line and does not cross it, the ridge does not produce a sign barrier on the line. Another projection line with an orientation substantially different from the orientation of the ridge (or, equivalently, from the that of the first projection line) will detect such a ridge, since the ridge parallel to the first projection line will appear perpendicular to the second line and cross it at some point, generating a sign barrier on it. For the projection lines to have sufficiently different orientations, two orthogonal lines would be the best choice.

After performing distance transformation, the distance map is generated as shown in Fig.2.

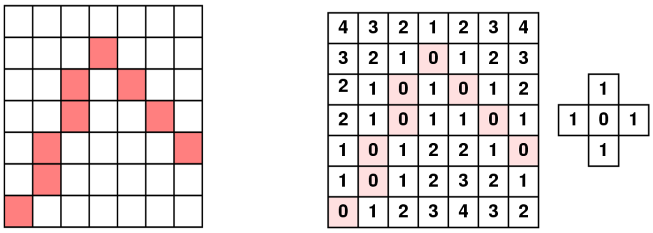


Fig. 2. Extracted feature points are marked by pink squares (left) and distance map using city block (or 4-neighbour) distance (right).

Voronoi diagram:

The Voronoi diagram of a discrete set of points (called generating points) is the partition of the given space into cells so that each cell contains exactly one generating point and the locus of all points which are nearer to this generating point than to other generating points as shown in Fig. 3.

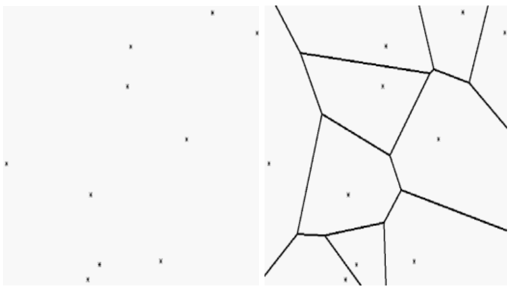


Fig.3. The 10 generating points (left) and their Voronoi diagram (right).

Using the Voronoi diagram to compute the skeleton of a polygonal shape is attractive because it results in skeletons which are connected while retaining Euclidean metrics Furthermore we obtain an exact medial axis compared to an approximation provided by other methods. Thus we may reconstruct exactly an original polygon from its skeleton invertibility one to one mapping. Finally algorithms to compute the Voronoi diagram and hence the skeleton are much faster than approaches that compute a distance transform.

Any method that utilizes Voronoi diagrams of polygons to compute skeletons must overcome the disadvantages listed below before it can be of practical value

-Natural shapes are non-polygonal. Thus accurate polygonal approximations of such shapes are required in order to compute skeletons without loss of accuracy.

-The skeleton of a many sided polygon of very short sides will have a large number of redundant edges because of the Voronoi edges at these vertices. This results in an increase in the complexity of the skeleton without significant addition of any shape information

-Finally robust and practical algorithms for Voronoi diagram construction of polygons are not very common

Most existing algorithms make assumptions about cocircularity of no more than three points and colinearity of no more than two. These constraints are difficult to satisfy in most practical applications

Basic Thinning Algorithms:

Thinning is a frequently used method for making an approximation to the skeleton in a topology- preserving way[9]. All thinning algorithms can be classified as one of two broad categories:

1. Iterative thinning algorithms and 2. Non-iterative thinning algorithms.

In general, iterative thinning algorithms perform pixel by-pixel operations until a suitable skeleton is obtained. Iterative algorithms may be further classified as[27]:

1) Sequential Thinning Algorithms

2) Parallel Thinning Algorithms

The above algorithms are shown in Fig. 4. Non-iterative thinning methods use methods other than sequential pixel scan of the image[6].

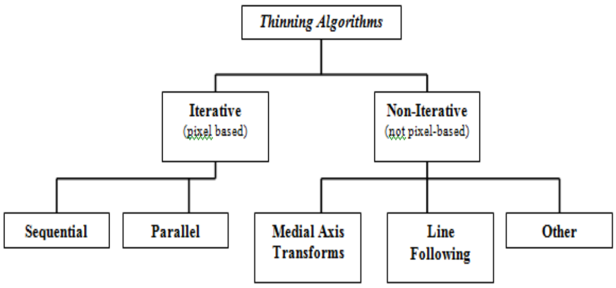


Fig. 4. Classification of Thinning Algorithms

The main difference between Sequential and Parallel thinning is that Sequential thinning operates on one pixel at a time and the operation depends on preceding processed results where as parallel thinning operates on all the pixels simultaneously. Sequential thinning produces better skeletons and requires less memories but parallel thinning is substantially faster. Most of the thinning algorithms developed were parallel algorithms.

Sequential Thinning:-

Using sequential algorithms, contour points are examined for deletion in a predetermined order, and this can be accomplished either by raster scan or contour following. Contour following algorithms can visit every border pixel of a simply connected object[1], and of multiply connected picture, if all the borders of picture and holes are followed[2]. These algorithms have an advantage over raster scans because they require examination of only the contour pixels instead of all pixels in P and in every iteration[12].

When a contour pixel p is examined, it is usually deleted or retained according to the configuration of N(p). To prevent sequentially eliminating an entire branch in one iteration, a sequential algorithm usually marks (or flags) the pixels to be deleted, and all marked pixels are then removed at the end of iteration. This generally ensures that only one layer of pixels would be removed in each cycle[1].

To avoid repetition, it is assumed that a pixel p considered for deletion satisfies all the following properties unless otherwise stated:

1. p is a black pixel.
2. p is not an isolated or end point, i.e., $b(p)\geq 2$.
3. p is a contour pixel, i.e., p has at least one white 4-neighbor.

Parallel Thinning:-

In parallel thinning, pixels are examined for deletion based on the results of only the previous iteration. For this reason, these algorithms are suitable for implementation on parallel processors where the pixels satisfying a set of conditions can be removed simultaneously. Divide each iteration into sub-iterations or sub-cycles in which only a subset of contour pixels are considered for removal[5][8].

At the end of each sub-iteration, the remaining image is updated for the next sub-iteration. Four sub-cycles have been used in which each type of contour point (north, east, south, and west) is removed in each sub-cycle. These have also been combined into two sub-iterations with one sub-iteration deleting the north and east contour points and the other deleting the rest[24].

In fast Parallel Thinning Algorithm, the method for extracting the skeleton of a picture consists of removing all the contour points of the picture except those points that belong to the skeleton[16][18]. In order to preserve the connectivity of skeleton, each iteration is divided into two sub-iterations[17].

p_9 (i-1,j-1)	p_2 (i-1,j)	p_3 (i-1,j+1)
p_8 (i,j-1)	p_1 (i, j)	p_4 (i,j+1)
p_7 (i+1,j-1)	p_6 (i+1,j	p_5 (i+1,j+1)

Table 1. Fast Parallel Algorithm for Thinning Digital Patterns.

According to Table 1, in the first sub iteration, the contour point p_1 is deleted from the digital pattern. If it satisfies following conditions[6]:

(a) $2 \leq B(p_1) \leq 6$

(b) $A(p_1) = 1$

(c) $p_2 \times p_4 \times p_6 = 0$

(d) $p_4 \times p_6 \times p_8 = 0$

In the second sub iteration, only condition (c) and (d)are changed and rest two conditions remain the same is shown below[6]:

(c') $p_2 \times p_4 \times p_8 = 0$

(d') $p_2 \times p_6 \times p_8 = 0$

where $A(p_1)$ is the number of 01 patterns in the ordered of $p_2, p_3, p_4, \ldots p_8, p_9$ that are the eight neighbors of p_1 and $B(p_1)$ is the non-zero neighbors of p_1, which is shown below:

$$B(p_1) = p_2 + p_3 + \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots + p_9.$$

If any condition is not satisfied then p_1 is not deleted from the picture. By condition (c) and (d) of the first sub-iteration it will be shown that the first sub iteration removes only the south-east boundary points and the north-west corner points which do not belong to an ideal skeleton[18].

	P2	
P8	P1	P4
	P6	

South

Fig. 5. Points under consideration and their locations.

Non-Iterative Thinning:-

These algorithm are considered to be non pixel based; they produce a certain median or center line of the pattern directly in one pass without examining all the individual pixels.

Some algorithms obtain approximations of skeletons by connecting strokes having certain orientations. For example, four pairs of window operations are used in four sub cycles to test for and determine the presence of right, or left diagonal, vertical, horizontal, limbs in the pattern. At the same time, the operators also locate turning points and end points by a set of final point conditions, and these extracted points are connected to form a line segment approximation of the skeleton. In [1], the boundary pixels are first labeled according to the above four local orientations. For each boundary pixel, a search is made for the same kind of label on the opposite side of the boundary (within a maximum stroke width) in the direction perpendicular to that given by the label.

Extraction of Medial Axis Transform:-

The medial axis transform (MAT) of an image is computed by calculating the Euclidean distance transform of the given input image pattern. The MAT is described as being the locus of the local maxima on the distance transform. After the computation of Euclidean distance transform (EDT) of the input image, the EDT is represented in image representing the Euclidean distances as gray levels.

The maximum Euclidean distance is represented as maximum gray level intensity in the EDT image. The pixel coordinates of the maximum gray level intensity are extracted from the EDT image by converting the EDT image into row x column matrix. The row and column of the matrix gives the coordinates of the MAT line of the image pattern.

III. TOPOLOGY PRESERVING SKELETONIZATION

In this section, a simple topology preserving skeletonization technique is proposed that iteratively removes pixels of a grayscale image. A grayscale image is taken as input. The output is a set of points that belongs to its skeleton. For every pixel in the grayscale image, check the acyclicity, i.e., if a candidate pixel is 1 and its 8-neighbourhood pixels are also 1, then it is cyclic else it is acyclic. If it is acyclic, then the value of that pixel is set to 1 otherwise 0. Check for connectivity to retrieve thinned component of the original image.

ALGORITHM: TOPOLOGY PRESERVING SKELETONIZATION

Require : Binary image b_i

Ensure : $S(b_i)$ that belong to the Skeleton of b_i;

Initially $S(b_i) = b_i$;
For every element $T \in S(b_i)$ do
if(bd(T)) is acyclic , then $S(b_i) = 1$;
while $S(b_i) \neq \emptyset$, do
For every element $T \in S(b_i)$ do
Compute $x = b_i - S(b_i)$;
[l, num] = bwlabel (x, 8);
initialize m=0;
if $b_i \neq x$, then
m=1;
if num $\neq 1$, then
$b_i = x$;
return $S(b_i)$;

IV. RESULTS AND DISCUSSIONS

The present method is applied on English alphabet set both on lower case and upper case as shown in Fig.3 and Fig.4. The English alphabets are chosen for experiment analysis because they contain different shapes. The thinned alphabet set contains no restoration and they are not affected by any border noises which are usually present in most of the skeleton approaches. One of the main problems with thinning method was loss of information due to binarisation because it could not always be possible to correctly binarise the whole character image using one threshold value.

Topology preserving skeletonization algorithm is proposed to evaluate its performance and examine its effects. The algorithms are chosen for representing different modes of operation in thinning [10]. The performance of the thinning algorithm can be compared and evaluated on the basis of following parameters:

a) Connectivity of Pattern.

b) Data Reduction Rate (DRR)

a) Connectivity of Pattern

By comparing the original image with its skeleton obtained by topology preserving skeletonization algorithm, we can see that the connectivity is maintained in the skeleton of the image.

b) Data Reduction Rate

The algorithm will guarantee the highest data reduction value producing perfect skeleton. It reveals information about how good is the algorithm in data reduction when comparing the skeleton S and the original pattern P. Formally: this can be measured as *Mdr*= |S|/|P|.

Data reduction rate for different images is shown in Table2.

SNO.	IMAGE	% OF REDUCTION
1	A	19.35
2	C	19.02
3	G	16.08
4	H	16.95
5	U	18.27
6	X	18.14
7	Y	17.86

Table 2. Data Reduction Rate for different images.

The present algorithm has been tested over all uppercase and lowercase alphabets. But for convenience, only some of the alphabets are shown in Fig.6 and Fig.7.

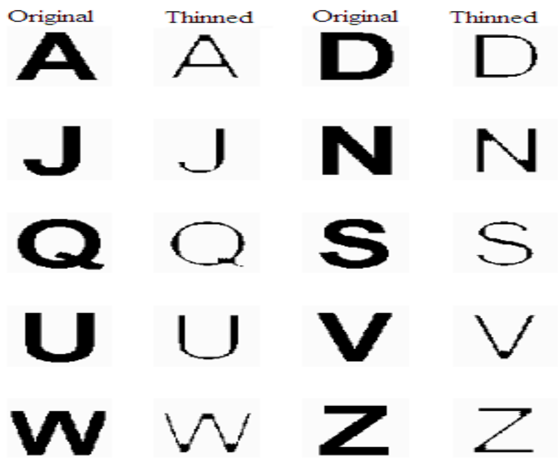
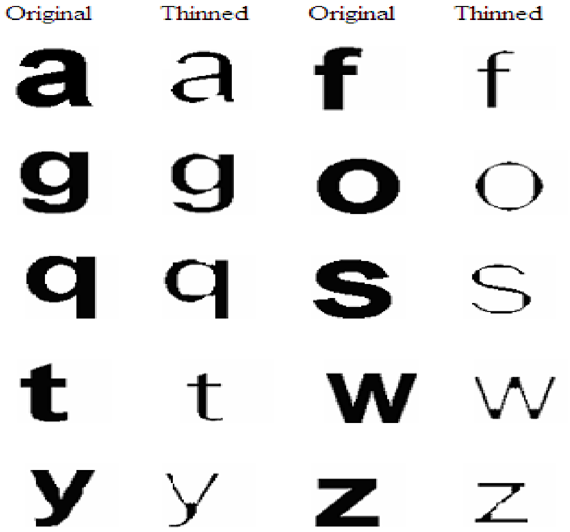


Fig.6. Original and Thinned Image of English Upper-case alphabets.



. Fig.7. Original and Thinned Image of English Lower-case alphabets.

IV. CONCLUSION

A topology preserving thinning algorithm based on iteratively culling pixels is introduced. The topology preserving thinning algorithm is used to retain the morphological properties of the images. It checks for the acyclicity condition and removes the pixels from a grayscale image. The shape preserving algorithm tends to retain the original shape of the image. This algorithm is very simple and easy to obtain skeletons. The performance of the skeleton is evaluated and is observed that for all the images connectivity is obtained and reduction rate is less than 20%.

REFERENCES

[1] L. Lam, S.-W. Lee, and C. Y. Suen, "Thinning methodologies—A comprehensive survey," IEEE Trans. Pattern Anal. Mach. Intell., vol. 14, no. 9, pp. 869–885, Sep. 1992.

[2] P. Wiederhold and S. Morales, "Thinning on cell complexes from polygonal tilings, Discrete Appl. Math.," vol. 157, no. 16, pp. 3424–3434, 2009.

[3] L. Liu, E. W. Chambers, D. Letscher, and T. Ju, "A simple and robust thinning algorithm on Cell complexes," 177, no. 5,pp. 975–988, 2001.

[4] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns", Comm. of the ACM 27, 1984, pp. 236-239 [5]. Zichang Guo and Richard W. Hall.

[5] "Parallel Thinning with Two- Subiteration Algorithms," Communications of ACM, vol. 32(3), March 1989, pp. 359-373.

[6] Harish Kumar and Paramjeet Kaur Deptt. Of CSA, Chaudhary Devi Lal University, Sirsa, Haryana, India1"A Comparative Study of Iterative Thinning Algorithms for BMP Images".

[7] W.H.Abdulla,A.O.M. Saleh and A.H. Morad, "A preprocessing algorithm for hand-written character recognition", Pattern Recognition Letters 7, Jan 1988, pp. 13-18.

[8] C. M. Holt, A. Stewart, M. Clint, and R. H. Perrott, "An improved parallel thinning algorithm," Commun. ACM, vol. 30, no. 2, pp. 156–160, 198.

[9] . K. Jang and R. T. Chin, "Analysis of thinning algorithms using mathematical morphology," IEEE Trans. Pattern Anal. Mach. Intell.,vol. 12, no. 6, pp. 541–551, Jun. 1990.

[10] Peter Tarabek, "Performance measurements of thinning algorithms", Journal of Information, Control and Management Systems, Vol. 6, (2008), No.2.

[11] A Datta and S. K. Parui, "A Robust Parallel Thinning Algorithm for Binary Images." Pattern Recognition, vol. 27, no. 9, March 1994, pp.1181-1192.

[12] M. Ashwin and G. D. Babu, "A new sequential thinning algorithm to preserve topology and geometry of the image," Int. J. Math. Trends Technol., vol. 2, no. 2, pp. 1–5, 2011.

[13] T.A. Aberra, "Topology Preserving Skeletonization of 2D and 3D binary images," M.S.thesis, Dept. Mat., Tech. Univ. Kaiserslautern, Germany, 2004.

[14] Jun-Sik Kwon, Jun-Woong Gi and Eung-Kwan Kang, 2001, "An Enhanced Thinning Algorithm Using Parallel Processing", IEEE, pp.no.752-755.

[15] Gabor Nemeth and Kalman Palagyi, 2009, "Parallel Thinning Algorithm Based on Ronse"s Sufficient Conditions for Topology Preservation", Research Publishing Service, pp.no.1-12, Aug.16.

[16] P. S. P. Wang and Y. Y. Zhang, 1986, "A fast serial and parallel thinning algorithm," in Proc. Eighth Euro. Meeting Cybern. Syst. Res. (Vienna, Austria), pp. 909-915.

[17] C. J. Ammann and A. G. Sartori-Angus, 1985), "Fast thinning algorithm for binary images," Image Ksion Comput., vol. 3, no. 2, pp. 71-79.

[18] "A comment on a fast parallel algorithm for thinning digital patterns," Comm. ACM, vol. 29, no. 3, pp. 239-242.1986.

[19] S. Suzuki and K. Abe, **1987),** "Binary picture thinning by an iterative parallel two- subcycle operation", Patt. Recogn., vol. 10, no. 3, pp. 297-307.

[20] M. P. Martinez-Perez, J. Jimenez, and J. L. Navalon, 1987, "A thinning algorithm based on contours", Compuf. Vision Graphics Image Processing, vol. 38, pp. 186-201.

[21] R. C. Gonzalez and P. Wintz, 1987, "The skeleton of a region," in Digital Image Processing. Reading, MA Addison-Wesley, pp. 398-402.

[22] Y. Y. Zhang and P. S. P. Wang, 1988), "A modified parallel thinning algorithm," in Proc. 9th Int. Con$ Putt. Recogn. (Rome, Italy), pp. 1023-1025.

[23] Richard W.Hall, 1989, "Fast Parallel Thinning Algorithms: Parallel Speed and Connectivity Preservation", Comm. CM. Jan, vol.32.no.1.pp.no.124-131.

[24] Y. Y. Zhang and P. S. P. Wang, 1996, "A Parallel thinning algorithm with two subiteration that generates One-Pixel-Wide Skeletons", Proc. IEEE Int"l Conf. Pattern Recognition, vol. 4, pp. 457-461.

[25] N.H.Han, C.W.La, and P.K.Rhee, 1997," An Efficient Fully Parallel Thinning Algorithm", IEEE -08186-7898-4, pp.no.137141.

[26] S. Yokoi, J. I. Toriwaki, and T. Fukumura, 1973, "Topological properties in digitized binary pictures," Syst. Comput. Contr., vol. 4, no. 6, pp. 32-39.

[27] Xingwei Yang, Xiang Bai, Xiojun and Wenyu Liu, 2008, "An Efficient Thinning Algorithm", IEEE, Computer Socity-2008, pp.no.475-478**.**

[28] P.S.P.Wang and Y.Y.Zhang, 1989, "A Fast and Flexible Thinning Algorithm", IEEE, Trans. May, Vol.38. No. 5. pp. 741-745.

AUTHORS

Dr. A. Sri Krishna received the PhD degree from JNTUK, Kakinada in 2010, M.Tech degree in Computer Science from Jawaharlal Nehru Technological University (JNTU) in 2003, M.S degree in software systems from Birla Institute of Technology and Science, Pilani in 1994, AMIE degree in Electronics & communication Engineering from Institution of Engineers, Kolkata in 1990. She has 23 years of teaching experience as Assistant Professor, Associate Professor, Professor and presently she is working as a Professor and Head, Dept of Information Technology at RVR&JC College of Engineering, Guntur. She has published 15 papers in International/ National Journals and Conferences. Her research interest includes Image Processing and Pattern Recognition. She is member of IE(I) and member of CSI.

Dr. K. Gangadhara Rao, Associate Professor in Computer Science and Engineering at Acharya Nagarjuna University, Guntur. His research interests include Computer Networks, Software Engineering, Object Oriented Anaysis and Design, Operating Systems, Cloud Computing.

N.Neelima received masters degree M.Tech(CSE) from Acharya Nagarjuna University in the year 2007, Guntur. She has 8 years of teaching experience. Presently she is working as an Assistant Professor in Department of IT, RVR & JC College of Engineering and she is pursuing Ph.D from JNTUH, Hyderabad. Her research interest includes Image and Signal Processing and computer vision. She is a member of IAENG.

K. Ratna Sahithi pursued her B.Tech Graduation in Information Technology in RVR & JC College of Engineering. She has been actively participating and presenting papers in student technical Symposium seminars at National Level. Her area of interest includes Image Processing, Computer Vision, Networks and Web Technology.