Detection and Correction of Page Orientation in Monochrome Textual Document Image

Vasudev .T[1] Hemantha Kumar .G[2] Nagabhushan .P[3]

Abstract

While imaging a document with a scanner, it is quite possible that the document is fed into the scanner in portrait or landscape or portrait flipped or landscape flipped directions, in other words the direction of feed could be 0° or 90° or 180° or 270°. Additionally, a document suffers a skew also. Hence, correction of directional orientation followed by skew correction becomes the first phase of processing in Document Image Analysis. In this paper, we provide a solution to these problems by employing a non rotational approach in two stages. A macro level skew correction(if exists) is made to the document image in portrait or landscape directions. The direction of orientation detection and correction is done in the later stage. The performance on English text documents is presented based on extensive experimentation.

Key words: Textual document image, page orientation, skew alignment, portrait and landscape directions, non-rotational approach

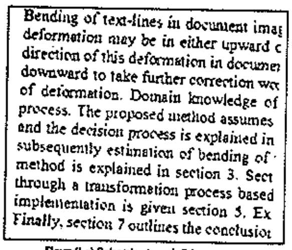
1. Introduction

Document Image Analysis(DIA) involves different stages starting from image acquisition to image understanding[1-3]. Each stage has many issues to be addressed and attempted to evolve generic solutions to the problems in DIA. Many researchers are attempting over these problems and the results are converging to build generic models. Some important issues in the image acquisition stage, are, (i) skew in image, the tilt given to the document while placing the document in scanner in its normal position, (ii) orientation in image, rotation in document image due to misfeeding of document to scanner in other directions to its normal position, and (iii) bending deformation in image, the bending effect due to scanning of bound document.

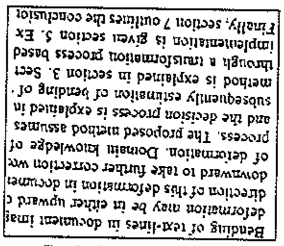
Considerable amount of research is reported in literature to the above. The number of works is quite high on skew detection, Jonathan and Taylor[4] in 1998 has made a survey on skew detection work and summarized over 40 approaches on skew detection. According to this survey, majority of methods are limited to detection of skew less than 30° and very few methods upto 45°. This is because in normal situation the skew will not be more than 15°. Each of the methods discussed[4] has its own limitations and advantages. Shivakumar[5] in 2005 has suggested different approaches to detect the quantum of skew in skewed documents. All these approaches end at finding the angle of skew and suggest to rotate the document image in the anti-direction of detected skew angle. Few works are reported in literature on detection and correction of deformation in image[6-8]. Zheng Zhang[6] in 2004, Breuel[7] in 2005 and Vasudev et. al. [8] in 2005 have made attempts on detection and correction of deformations in document images.

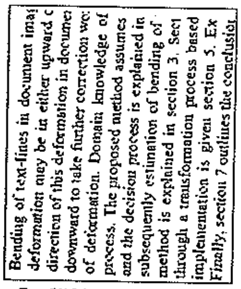
[123]Department of Studies in Computer Science, University of Mysore Manasagangotri, Mysore, Karnataka, India – 570 006. E-mail : banglivasu@yahoo.com

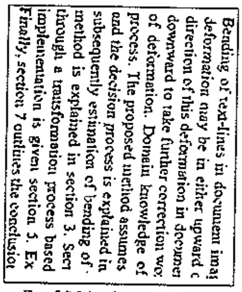
The direction of a document image is classified broadly into two, portrait and landscape. The document image is in portrait direction when it appears in its normal position or upside down as shown in figs 1(a) and 1(c). The document image is in landscape direction when it appears in horizontal position as shown in figs 1(b) and 1(d). Orientation in an image is due to the direction of placing the document into the scanner. During the process of scanning, there is quite possibility to feed the document to scanner by placing the document in different directions. Feeding of documents in different directions to scanner introduces corresponding orientations in the image. Feeding the document in portrait direction introduces an orientation in image close to 0^0 or 180^0. Similarly, feeding the document in landscape direction introduces an orientation close to 90^0 or 270^0. Figures (1a), (1b), (1c) and (1d) show the orientation of images close to 0^0, 90^0, 180^0 and 270^0 respectively. Further, skew/tilt is the small amount of rotation introduced with in a direction. Such oriented images with skew are not suitable for further processing.


Bending of text-lines in document imag
deformation may be in either upward c
direction of this deformation in documen
downward to take further correction wo
of deformation. Domain knowledge of
process. The proposed method assumes
and the decision process is explained in
subsequently estimation of bending of
method is explained in section 3. Sect
through a transformation process based
implementation is given section 5. Ex
Finally, section 7 outlines the conclusion

Figure(1.a) Orientation towards 0 degree (Portrait - Normal)


Figure(1.c) Orientation towards 180 degree (Portrait - Flipped)


Figure(1.b) Orientation towards 90 degree (Landscape - Normal)

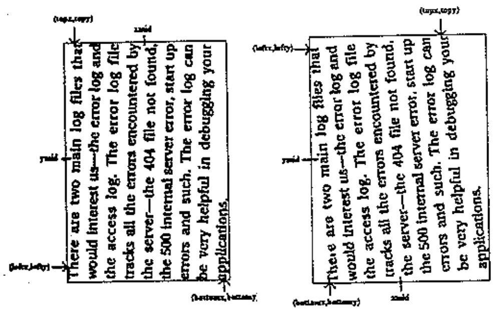

Figure(1.d) Orientation towards 270 degree (Landscape - Flipped)

Detection of page orientation would be simple task if the layout of the document is known apriori. In many general cases, such knowledge will not be available and detection of page orientation requires extraction of certain global features from the document. In this direction researchers have made attempts on detection of page orientation and the same is reported in literature. A methodology is proposed by Aditya, Zhang et.al[9]. in 2002 to detect orientation in non textual images. They have adopted Bayesian classifier for estimating the orientation and method is not extendable to text images. D.X.Le, G.Thoma, H.Weschler[10] have proposed an algorithm to detect page orientation and skew in document using projection profile in 1994. The limitation of the methodology is that it can detect the orientation of the document in portrait and landscape direction but do not distinguish between normal and flipped with in portrait or landscape direction. Avila and Lins[11] in 2005 suggested a fast method to detect skew and orientation in text images. The method is limited only to detect skew between 90° to -90° to normal position but do not address the detection of flipped orientations. In 2000, Caprari[12] had come out with a model for detection of page up/down orientation. This approach is limited to detect only orientations in 0^0 and 180^0.

Majority of the methods reported in literature are developed assuming that the document is in perfect landscape or portrait position. After detecting the orientation, the correction work is left unaddressed or suggested to adopt rotation transformation. Most of the work published on orientation and skew discuss the issues individually and have not combined both of them. This motivated us to come out with a model that performs all the three operations, (i) correct the skew with in a direction, (ii) detect the orientation and, (iii) correct the orientation through a non-rotational approach. Normally, while feeding a document the sequence of actions which cause erroneous image is – direction of placing the document followed by skew. But, in our work we correct the skew first and orientation correction is followed later. This sequence of actions simplifies the task and same is discussed later. The proposed work is presented in the subsequent sections. Section 2 describes the method used to detect and correct the skew in the document image with respect to the direction of landscape or portrait. Section 3 describes the method adopted to detect and correct the orientation of the document image. In both skew and orientation correction process a non-rotational approach is used. The experimental results are discussed in section 4 and section 5 outlines the conclusion on the proposed work.

2. Detection and correction of skew with in portrait and landscape direction

This work is a modification of our earlier work[13], developed to detect and correct skew in language independent printed documents. Initially it is required to detect the nature of skew ie., document is skewed to left or right. A macro level decision is made to identify the nature of skew based on the boundary parameter values. A rectangular boundary is imagined surrounding the skewed document as shown in fig(2). *xmid* and *ymid* are x and y coordinates of mid points on the boundary lines. *topx* and *topy* are x and y coordinates indicating the top most point in the documents. *leftx* and *lefty* are x and y coordinates indicating the left most point in the document image. Similarly, *bottomx* and *bottomy* are coordinates indicating bottom most point in the document image.



Fig(2) Skewed image surrounded by box with parameters

It is quite possible to make a macro level decision from the fig(2) and domain knowledge of printed documents as follows,

If $topx < xmid$ then

the document is skewed **clock-wise**

If $topx > xmid$ then

the document is skewed **counter clock-wise**

Next, it is required to estimate a left boundary line for the skewed document and this is initially approximated between the points (*topx,topy*) and (*leftx,bottomy*) for clock-wise skewed documents and between the points (*leftx,topy*) and (*bottomx,bottomy*) for counter clock-wise skewed documents. As this initial approximation is not always the correct left boundary line a re-approximation is made through searching a suitable line that closely forms the left boundary line for the skewed document as illustrated in fig(2). For the sake of comprehension the algorithm for fixing the left boundary line is given for clock-wise skewed document.

In a clock-wise skewed document, it is required to approximate the left boundary line between the points (*topx,topy*) and (*leftx,bottomy*). Since no other point of the document exists to the left of *leftx*, the point (*leftx,bottomy*) is fixed and only the point at top (*topx,topy*) is varied for re-approximating the left boundary line. The function drawline() is a DDA line drawing algorithm[14] which draws a line between two specified points. The algorithm for finding the left boundary line is as follows.

1. drawline(*leftx,bottomy,topx,topy*)

2. while (line intersects any character of text) do

a. $topx = topx - 1$

b. drawline(*leftx,bottomy,topx,topy*)

The final line that is obtained from the algorithm is the re-approximated left boundary line of the skewed document and is extended till the boundary as shown in fig(3a). Similarly, initial and final left boundary lines are drawn starting from points (*leftx,topy*) and (*bottomx,bottomy*) for counter clock-wise skewed document. The left boundary line is the seed line to approximate the other boundary lines of the skewed document.

The estimated left boundary line '*L1*' of the document is the seed line to draw other boundary lines of the document is shown in Fig 3(a). An initial right boundary line '*L2*' is obtained as parallel line to '*L1*' at right edge of the document. The final position of Line '*L2*' is obtained by sliding line '*L2*' from the initial position towards left until the line '*L2*' just touches the document from right side. Initial and final positions of line '*L2*' are shown in fig(3b). The top line '*L3*' perpendicular to '*L1*' is obtained based on the following procedure.

P1 is the top point on '*L1*'

S1 is the slope of '*L1*'

P2 is the top point on '*L2*'

Draw an initial top line '*L3*' between points *P1* and *P2*

S3 is the slope of '*L3*'

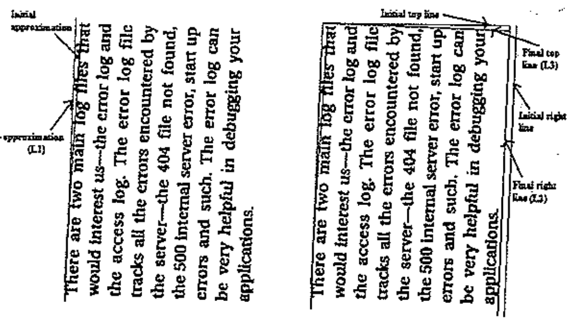
While ($S1 * S3$ not with in the range[-1.1 to -0.9]) do

P2 is the next point on '*L2*'

Draw a new line '*L3*' between points *P1* and *P2*

Find slope *S3* of new line '*L3*'

Final line '*L3*' obtained is the top boundary line of the document Initial and final positions of line '*L3*' is shown in fig (3b).

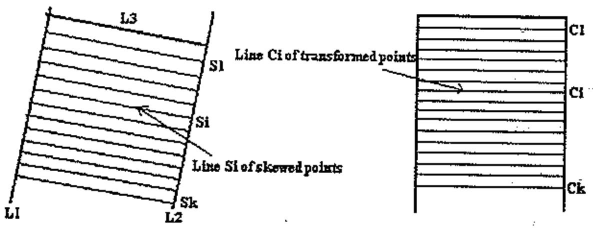


Fig(3a): Estimation of left boundary line

Fig(3b): Fixing right and top boundary lines

The three bounding lines '*L1*', '*L2*' and '*L3*' bounds the text area of the skewed image and is required to be corrected through a transformation process for further processing. A non-rotational approach[13] is adopted to correct the skew and the approach is defined below.

The transformation process collects all the points of the skewed document on a line formed between left and right boundary line parallel to top boundary line and place them on a horizontal line. This transformation results the skewed points repositioned into appropriate position. For comprehension, a set of skewed lines are shown as skew corrected lines using the approach in fig(4).



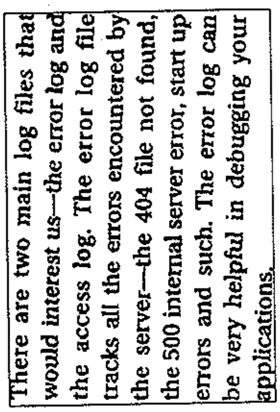
Fig(4): Transformation of skewed points to correct position

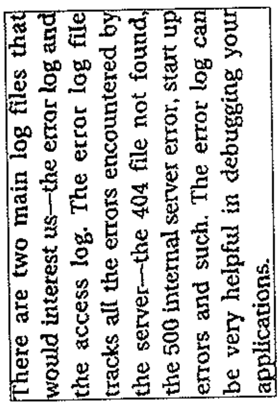
Lines S1...Si...Sk are the are the possible lines of pixels between '*L1*' and '*L2*' which are parallel to '*L3*'. Lines C1...Ci...Ck are the transformed lines of S1...Si...Sk respectively and m is the number of points in each line S. The transformation S to C is given as follows, For all j=1…k

$S_j=P_j$, where P_j is a set points $(p_{j1} \cdots p_{jm})$of skewed j^{th} line S_j

Cj= set of points P_j $(p_{j1} \cdots p_{jm})$ placed on the j^{th} horizontal line C_j.

The result of the skew correction process is shown in fig(5b) for the skewed input document in landscape direction shown in fig(5a).


There are two main log files that would interest us—the error log and the access log. The error log file tracks all the errors encountered by the server—the 404 file not found, the 500 internal server error, start up errors and such. The error log can be very helpful in debugging your applications.


There are two main log files that would interest us—the error log and the access log. The error log file tracks all the errors encountered by the server—the 404 file not found, the 500 internal server error, start up errors and such. The error log can be very helpful in debugging your applications.

Fig(5a): Input skewed document

Fig(5b): Skew corrected document

3. Detection and correction of orientation

A simple x-cut and y-cut technique[1-3] is adopted to find the document is in portrait or landscape direction. First, the document is scanned along x axis from top to bottom. If this scanning leads to segmentation of lines in the document then the document is in portrait direction (0° or 180°). In case, if line segmentation is not possible in y direction, then the document is scanned along y axis from left to right. If the scanning leads to segmentation of lines then the document is in landscape direction (90° or 270°). Even if this scanning fails to segment the lines of document in x direction, then a failure case is reported.

Further, based on the decision made on the direction of the document, the document is arbitrarily divided into five parts along the text lines as shown in fig(6a) through fig(6e). A logical OR operation is performed on theses five parts of the document and the result of the operation is shown in fig(6f). The division and OR operations are performed to obtain a better average pixel distribution over the lines of text and at the same time computational size also gets reduced.

The pixel distribution graph for fig(6f) is shown in fig(7). Each segment in the graph represents a text line of the document. The central region bound by two peaks with in a segment is the centre area of a text line. Remaining two regions on either side of the centre region corresponds to upper region and lower region of the text line. The domain knowledge of printed English text states, that each line is divided into three regions namely, upper, middle and lower regions. The middle region normally has relatively uniform distribution of pixel density and has maximum value when compared with upper and lower regions pixel densities, whereas the upper region has higher pixel distribution density than the lower region. Based on this domain knowledge about distribution of pixels in these two regions, a decision is made to find the orientation of the document. The orientation of the document is decided based on the procedure specified below.

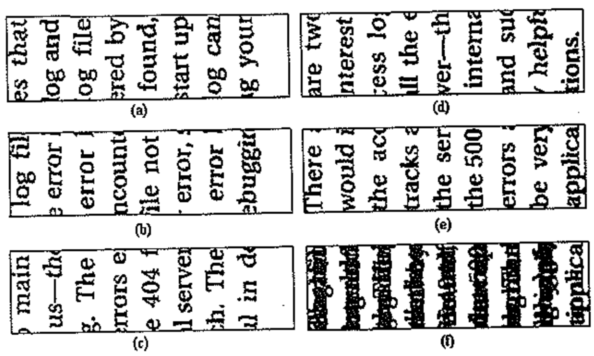
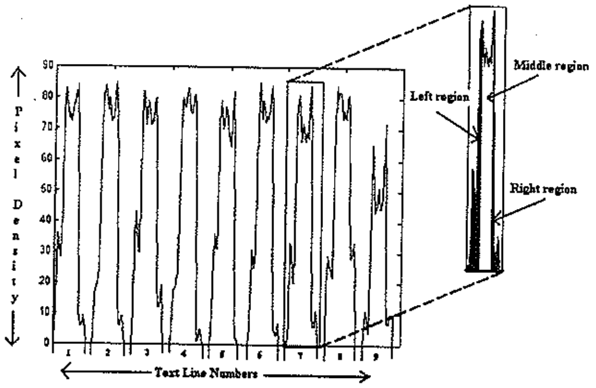


Fig6(a-e): Five parts of the document, Fig(6f) : Logical OR of five parts

Let $dleft_i$ be the density of pixels in the region to the left of central region and $dright_i$ be the density of pixels in the region to the right of central region on i^{th} segment. Let k be the number line segments in the graph. Left region density(DL) and right region density(DR) are given by the following equations.



Fig(7): Pixel density distribution graph of the fig(6f)

$$DL = \sum_{i=1}^{k} dleft_i \quad \text{and} \quad DR = \sum_{i=1}^{k} dright_i$$

Average $DL = DL / k$ and Average $DR = DR / k$

For a document in portrait / landscape direction,

If Average $DL >$ Average DR then

Document is oriented by $0^0/ 90^0$

Else

If Average $DL <$ Average DR then

Document is oriented by $180^0 / 270^0$

Else

Failure

The computations performed for detection of orientation is made on the logically 'OR'ed image fig(6.f) using point processing technique[2,3]. The computational complexity is slightly high when this approach is adopted. In order to reduce the computational complexity, the pixels distribution density is straight away computed from the divided parts of the image. The pixel distribution density result of the later approach is same as that of the earlier approach as shown in fig(7). Analysis of computational complexities of both the approaches is discussed below assuming the size of the divided images as n x n.

The logical 'OR'ing approach requires, four times 'OR' operations in 'OR'ing five sub images each with complexity $O(n^2)$ and one histogram operation of complexity $O(n^2)$. A total of 5 times $O(n^2)$ operations are required in this approach. The direct computation of histogram approach requires, one histogram computation by 'OR'ing the pixel position from five sub images with in a single loop with time complexity $O(n^2)$. A total of 1 time $O(n^2)$ operations are noticed in this approach and this approach of obtaining histogram directly from sub images is comparatively 20% efficient than obtaining the histogram from 'OR'ed image of sub images.

The sequence of actions – skew correction followed by orientation correction plays important role in this work. The histogram or the pixel density distribution can be obtained properly, only when the skew correction is done otherwise, it is not possible to obtain the histogram that provides sufficient information to make the decision about the orientation.

After detecting the orientation, the correction process is followed and the correction of orientation is performed based on the same principle of non-rotational approach described in section 3 for skew alignment of the document. Accordingly, for a 90^0 oriented document, all the vertical pixel lines from left to right of document are transformed as horizontal pixel lines from top to bottom of resultant document. For a 180^0 oriented document, all the horizontal pixel lines from bottom to top are transformed as horizontal pixel lines from top to bottom of resultant document. For a 270^0 oriented document, all the vertical pixel lines from right to left are transformed as horizontal pixel lines from top to bottom of resultant document. The results of the experiments are discussed in next section.

4. Experimental Results

The developed method has been tested over more than 600 samples of printed English documents. The samples are obtained with different resolutions, different skew angles, and different orientations. Table(1) gives the summary of results of skew correction process and table(2) gives the summary of the orientation correction process.

Table(1): Results of skew correction process

Skew direction	No. of samples	Result without error	Result with error
0^0	154	139(90.25%)	15(09.75%)
90^0	157	145(92.36%)	12(07.64%)
180^0	158	142(89.87%)	16(10.13%)
270^0	155	140(90.32%)	15(09.68%)
Total	624	566(90.71%)	58(09.29%)

Table(1) indicates an average efficiency of 90.71% in skew correction stage without error. Around 9.29% is reported as skew correction with error. The error in this stage is due to the error introduced in estimation of left boundary line for the skewed document. No failure cases are reported in this stage.

Table(2): Results of Orientation detection process

Orientation direction	No. of samples	Detected	Rejected	Failure
0^0	154	143(92.86%)	6(03.90%)	5(3.24%)
90^0	157	147(93.63%)	6(03.82%)	4(2.55%)
180^0	158	144(91.14%)	7(04.43%)	7(4.43%)
270^0	155	142(91.61%)	8(05.16%)	5(3.23%)
Total	624	576(92.31%)	27(04.33%)	21(3.36%)

Table(2) shows an average efficiency of 92.31% of orientation detection with 4.33% of rejection and 3.36% of failures. Rejections are due to larger error in skew correction process. If the skew correction process results with an error $>0.4^0$ then the method rejects the case. Failures are due to smaller error in skew correction process. If the skew correction process results with an error in the range $>0.2^0$ and $< 0.4^0$ then there is a possibility in the method to misclassify the orientation. The skew correction process results with an error $<0.2^0$ yields correct detection of orientation. The overall efficiency of the method is 91.51%. The method is tested with only English text documents and results are not good enough for Kannada and other South Indian script documents. However, the skew correction stage of this work is extendable to other languages[13].

5. Conclusion

The described method for automation of orientation detection and correction is a two stage process. In the first stage, the skew with in the portrait or landscape of the document image is detected and corrected. This stage shows an efficiency of 90.71%. The second stage detects the orientation of the document and corrects the same. This stage shows an efficiency of 92.31%. In both the stages, a non-rotational approach is adopted for correction process. The developed method shows an overall efficiency of 91.51%. The reason for failure and rejection is mainly due to the error in skew correction process. The error in skew correction is introduced due to error is estimating the left (seed) boundary line of the document. A reduction of error in estimating the left boundary line improves the efficiency of the method. The computational complexity of the method is $O(n^2)$ as both the stages use point processing technique[2,3] assuming the size of the image as n x n. Any image processing application with computational complexity as O(n^2) is considered to be relatively less expensive. The method is limited to English text documents without tables and pictures.

References

[1] Lawrence O'Gorman, Rangachar Kasturi, *"Executive Briefing: Document Image Analysis"*, IEEE Computer Society Press, 1998.

[2] A.K.Jain, *"Fundamentals of Digital Image Processing"*, Prentice Hall, Englewood Cliffs, NJ, 1989.

[3] Rafael C Gonzales & Richard E Woods, *"Digital Image Processing"*, Addison Wesley Publication, 1987.

[4] Jonathan J Hull, S.L.Taylor, *"Document Image Skew Detection: Survey and Annotated Bibliography"*, World Scientific, PP 40-64, 1998.

[5] Shivakumar P, *"Generation of Complete Large Document Images from Split Components"*, Ph.D thesis under the supervision of Hemanthakumar G, University of Mysore, India, 2005.

[6] Zheng Zhang, *"Restoration of Curved Document Images through 3D Shape Modeling"*, In International Conference on Computer Vision and Pattern Recognition (CVPR2004), June 2004.

[7] T.Breuel, *"The Future of Document Imaging in the Era of Electronic Documents"*, Proc. International Workshop on Document analysis, India, PP 275-296, 2005.

[8] Vasudev T, Hemantha kumar G, Nagabhushan, *"An Elliptical Approximation Model for Removal of Text-line Bending Deformations at Page Borders in a Document Image"*, Proc. International Conference on Cognition and Recognition, (ICCR2005), India, PP 645-654, 2005.

[9] Aditya Vailaya, HongJaing Zhang, Changjiang Yang, Feng-I Liu, Anil K Jain, *"Automatic Image Orientation Detection"*, IEEE Transactions on Image Processing, Vol. 11, No. 7, 2002.

[10] D.X.Le, G.Thoma, H.Weschler, *"Automated Page Orientation and Skew Angle Detection of Binary Document Images"*, Journal of Pattern Recognition, Vol. 27, No. 10, PP 1325-1344, 1994.

[11] B.T.Avila, R.D.Lins, *"A Fast Orientation and Skew Detection Algorithm for Monochromatic Document Images"*, ACM Symposium on Document Engineering, PP 302-316, 2005.

[12] R.S.Caprari, *"Algorithm for Text Page Up/Down Orientation Determination"*, Pattern Recognition Letters, Vol. 21, No. 4, PP 311-317, 2000.

[13] Murali.S, Vasudev.T, Hemanthkumar.G, Nagabhushan.P, *"Language Independent Skew Detection and Correction of Printed Text Document Images: A Non-rotational Approach"*, Vivek – Journal of Artificial Intelligence, Vol. 16, No. 2, PP 8-15, 2006.

[14] Donald Hearn, M.P.Baker, *"Computer Graphics"*, Pearson Education, 2nd Edition, 2003.

Author's Biography

Vasudev T is working as Assistant Professor and Head of Department of Computer Applications, P.E.S College of Engineering, Mandya, Karnataka. He is working towards his Ph.D in the area of Document Image Analysis(DIA) under the supervisions of Dr.G.Hemanth Kumar and Dr.P.Nagabhushan, Department of Studies in Computer Science, University of Mysore, Mysore, Karnataka. His areas of specialization and interest are Digital Image Processing, Pattern Recognition, Database Management Systems and Data Structures.

Dr. Hemanth Kumar G is working as Professor, Department of Studies in Computer Science, University of Mysore, Mysore, Karnataka. His areas of interest are Digital Image Processing, Pattern Recognition, Document Image Analysis, Epigraphical Document Image Processing, Speech Processing, Face Recognition. Under his supervision a total of 5 Doctoral degrees are awarded, 2 candidates have submitted their thesis for evaluation and another 4 candidates are still working. He has nearly 200 publications in journals and conferences.

Dr. Nagabhushan P is Professor and Chairman, Department of Studies in Computer Science, University of Mysore, Mysore, Karnataka. His areas of interest are Pattern Recognition, Digital Image processing, Remote Data Analysis, Data Clustering, Analysis and Design of Algorithms. So far a total of 13 Doctoral degrees are awarded under his supervision and another 5 Ph.D students are working under his supervision. He has around 500 publications in journals and conferences.