Chapter 2

Preparations on images of Input Forms: A Document Image level Pre-processing for Automatic Data Extraction

There are many types of documents; one is that of questionnaire type. A user should fill up the various questions/enquiries which are apriori/preprinted on the document. Such documents are generally called FORMS. Specifically they are given various names, such as application forms, challan, data forms, exam evaluation forms, leave forms and so on. Naturally preprinted portions pose some questions/enquiries to the user, and an user responds by filling up the answers in the space provided. The preprinted questions are obviously the INVARIANT aspect of the document, and the answers/responses filled up in the space earmarked for the purpose represent the VARIANT portions, because these are different over different users. In this research work the automation required is to go to the right VARIANT blocks and pick up the contents from these variant blocks. However before this could be done, what should essentially be carried out is to recognize the type of form that is fed into the input. The basic problem in the feed operation is that the input form suffers two major dis-lodged placements viz.,

i. portrait/landscape possible with flips
ii. a slight tilt called skew

Unless these dis-position problems are overcome, the form fed-in cannot be recognized. Hence the first preparatory task to be undertaken is to set right the dislodged placements of images of input forms.

In this research work two new preprocessing approaches are proposed. The first method is to detect and correct the skew within portrait/landscape direction and the second method is to detect and correct the orientation in the images of documents. The rest of the chapter is organized as follows: Section 2.1 gives a brief background about
the requirement for preparations on images of input forms. Section 2.2 explains the method proposed for skew detection and correction of input forms with experimental illustrations. In section 2.3 the method developed for detection and correction of orientation is explained with experimental illustrations. The methods proposed for automatic detection and correction of orientation in document images are not suitable for extending the same for images of input forms. Specific methods are required for detection and correction of orientation in the images of input forms. The methods developed based on specific type of input forms is explained in section 2.4 with instances. Section 2.5 briefs the summary of this chapter.

2.1 Background

The efficiency of the methods proposed in literature about the various activities involved in data extraction process like document type identification, detection of variant regions, recognition of characters are sensitive to rotation in images[1]. The methods reported in literature exhibit lower efficiencies if the images are not found intact, in other words if the image suffer rotation. The reductions in performances of the reported methods demand the input form images to be free from rotation. In order to widen this requirement, a preprocessing stage is essential to position automatically the resulting image to be free from rotation. Obtaining the input form image free from rotation for automatic data extraction essentially requires two operations; Orientation correction and Skew correction.

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 figures (2.1a) and (2.1b). The document image is in landscape direction when it appears in horizontal position as shown in figures (2.1c) and (2.1d). Orientation in an image is due to the direction of placing the document into the scanner. During the process of scanning, there is always possible 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 $0^\circ$ or $180^\circ$. Similarly, feeding the document in landscape direction introduces an orientation close to $90^\circ$ or $270^\circ$. Figures (2.1a), (2.1b), (2.1c) and (2.1d) show the orientation of images close to $0^\circ$, $90^\circ$, $180^\circ$, and $270^\circ$ respectively. Further, skew/tilt is the small amount of rotation introduced with in a direction. Figure(2.1e) shows a skewed image in $0^\circ$ direction and figure(2.1f) shows a skewed image in $270^\circ$ orientation.

The document images shown in figures (2.1b) through (2.1f) are not suitable for automatic reading by Optical Character Readers (OCRs) as these images are rotated and OCRs performances are sensitive to rotations. In addition, the positions of variant regions in the input form also get disturbed in rotated document images due to changes in the size of image. Hence, some preparatory tasks are to be performed initially on such rotated and skewed document images before subjecting them for automatic reading. The initial preparation task requires two sequence of operations (i) removal of skew within an orientation and (ii) detection of orientation and positioning the document image into proper orientation($0^\circ$). The method proposed for skew angle correction within an orientation is explained in the next section.

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. The access log tracks all the errors encountered by the server—the 404 file not found, the 500 internal server error, start up errors and such. The access log file would interest us—the error log and the access log.
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.
2.2 Skew Correction in document images within an orientation

Skew in image is the tilt/inclination/angular rotation given to the document while placing the document in scanner in its normal position[Gon102; Gor98]. But introduction of skew in images is possible in any of the $0^0$, $90^0$, $180^0$ and $270^0$ orientations. First, it is necessary to correct the skew in the image within an orientation and next, the detection and correction of orientation in images is performed. In this direction a model has been developed to detect and correct the skew in printed document images within an orientation using a non-rotational approach. For the sake of comprehension, the method for detection and correction of skew is explained by considering a document image in $0^0$ orientation, and the method can be used for skew detection and correction in other orientations. Initially, the skew correction of general documents is considered and then the method is used for skew correction of input forms.

Considerable amount of research is reported in literature on the skew detection[Gat97; Ami100; Pal101; Kav102; Lio103; Yue103; Shi105]. It is evident from literature that the majority of these methods mainly focus on skew angle detection in document images. In 2000, Amin and Fischer have come out with a skew detection in document images using Hough transformation[Ami100]. Pal, Mitra and Choudhuri in 2001 have worked on detection of multi skewed text lines in documents[Pal101] and suggested existing skew correction approaches. Kavallieratou, Fakotakis and Kokkinakis in 2002 used Wigner-Ville distributions to estimate skew angle in document images[Kav102]. Yang Cao, Shuhua Wang and Hang Li have developed skew detection in document images based on straight line fitting[Yue103] in 2003.

Shivakumar[Shi105] in 2005 has suggested different approaches to detect the quantum of skew in skewed document images. These research works are converging towards accurate skew angle detection in images and the skew correction is left unaddressed suggesting to adopt the rotation transformation[Donl03]. The rotation transformation approach requires the angle of skew in order to correct the skew through rotating the image in counter direction of skew.

The result of the skew correction processes depends on the accuracy of the skew angle detected in images. The limitation in finding the accurate skew angle and then adoption of the correction process has motivated us to investigate an approach to perform skew correction without looking into the contents of the documents and the angle of skew. A non-rotational transformation has been proposed for skew correction that produces results comparable to other approaches.

The work presented in this section focuses on detection and correction of skew in language independent printed text document images ranging from -30° to +30° in any orientation. A method is developed to detect the skew in printed document image and a non-rotational transformation process is proposed to correct the skew. The method uses linear approximation approach to correct the skew directly without finding the skew angle. The method consists of two stages. In the first stage, the direction of skew is detected based on fixing boundary lines across the corners of the document image. The second stage corrects the skew in the document image. The proposed approach for skew detection and correction in document images exploit line drawing algorithm[Don103]. The method has been tested on varieties of printed document images with different skew angles and scripts.

The approach assumes that no part of document is missing in the image i.e, image of the complete document is available, noise free and atleast 70% of text lines are uniformly aligned on the boundary lines. The process of an initial approximation and re-approximation on direction of skew and fixing the left boundary line for the text is explained in subsection 2.2.1. The process of fixing right boundary and top boundary is explained in subsection 2.2.2. The process of non rotational transformation is given
in subsection 2.2.3. Experimental results and failure analysis is summarized in subsection 2.2.4. The readability analysis of the skew corrected document images is discussed in subsection 2.2.5. Conclusion on the work is discussed in subsection 2.2.6.

### 2.2.1 Left boundary line approximation

In order to proceed with skew correction process, initially it is required to find the nature of skew, i.e., determining the document is skewed clock-wise or counter-clock-wise. A domain knowledge 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 figure(2.2.1a). $x_{mid}$ and $y_{mid}$ 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.

![Diagram](image)

Figure(2.2.1a) Skewed image surrounded by box with parameters

It is quite possible to make a decision from the figure(2.2.1a) and domain knowledge of printed documents as follows,
If \( topx < xmid \) and \( lefty > ymid \) then
the document is skewed \textbf{clock-wise}
If \( topx > xmid \) and \( leftx < ymid \) then
the document is skewed \textbf{counter clock-wise}

It is required to estimate a left boundary line for the skewed document and this is
initially approximated between the points \((topx, topy)\) and \((leftx, lefty)\) for clock-wise
skewed documents and between the points \((leftx, lefty)\) 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 exactly forms the left boundary line for the skewed document as illustrated in
figure(2.2.1b). For the sake of comprehension the explanation and algorithm for fixing
the left boundary line is given below for clock-wise skewed document.

In a clock-wise skewed document, it is required to fix the left boundary line between
the points \((topx, topy)\) and \((leftx, lefty)\). Since no other point of the document exists to
the left of \( leftx \), the point \((leftx, lefty)\) is fixed and only the point at top \((topx, topy)\) is
varied. The function \texttt{drawline()} is a DDA line drawing algorithm\[Donl03\] which
draws a line between two specified points. The algorithm for finding the left boundary
line is as follows.

1. \texttt{Drawline(leftx, lefty, topx, topy)}
2. While (line touches any character of text) do
   a. \( topx = topx - 1 \)
   b. \texttt{drawline(leftx, lefty, 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
figure(2.2.1b). Similarly, initial and final left boundary lines are drawn starting from
points \((leftx, lefty)\) and \((bottomx, bottomy)\) for counter clock-wise skewed document.
The left boundary line is the seed line to approximate the right and top boundary lines
and the process is explained in next section.
2.2.2 Approximation of right and top boundary line

The right boundary line is approximated by sliding a line starting from the left boundary line over the top and bottom edge of the image towards right. The sliding is stopped when the resulting line does not have any character of the text to its right as shown in figure(2.2.2a) and this forms the right boundary line for the document. Decision on finding a part of text to the right of slided line is made on finding the occurrence of a point of a character on the line or in the remaining area of document towards the right end of the document image. Left and right boundary lines are referred as line1 and line2. (x1,y1) and (x2,y2) are the top and bottom points of line1. With these parameters the algorithm for approximating right boundary line is as follows.

1. Drawline(x1,y1,x2,y2)
2. While(there exists some part of text to the right of line)
   c. x1=x1+1
   d. x2=x2+1
   e. Drawline(x1,y1,x2,y2)

The final line that is obtained is the approximated right boundary line of the skewed document image.
Further, it is required to approximate the top boundary line. This boundary line should be perpendicular to both line1 and line2. In order to obtain the top boundary line for counter clock-wise skewed document image, we proceed from the top points (x1,y1) of line1 and (x2,y2) of line2 as shown in figure(2.2.2b). Keeping (x2,y2) as fixed, a point (x1',y1') is searched on line1 starting from (x1,y1) such that a line formed between (x1',y1') and (x2,y2) is perpendicular to line1 and line2. This new line formed is labeled as line3 is the approximated top boundary line as shown in figure(2.2.2b). Similarly, for clock-wise skewed documents, keeping (x1,y1) as fixed, a point (x2',y2') is searched on line2 starting from (x2,y2) and line line3 between (x1,y1) and (x2',y2') is the top boundary line as shown in figure(2.2.2b). Some buffer space is available in the top region, however this can be removed by sliding the line3 until it touches the top line of the text.
Statistical analysis from a corpus of Bangla text is useful and important in designing the OCR apart from other applications in Natural Language Processing and Cryptography. Since no such analysis is available in the literature, we collected two sets of data, the first set containing 100,000 words from popular Bangla books, newspapers and juvenile literature and the second set containing 60,000 words from a Bangla dictionary. Each datum is a word converted into ASCII codes of Fig. 1. A typical example is /NT/ RGH /A, TMU, LK. Here, /NT/ denotes that N and T together make a compound character. The words in the data sets were stored in alphabetical order. For the first set of data each distinct word is stored with its frequency of occurrence.

Algorithm for approximating the top boundary line for clock-wise skewed document is given below.

1. $s_1 =$ slope of $l_1$
2. $s_2 =$ slope of $l_2$
3. $(x_2', y_2') = (x_2, y_2)$
4. $s_3 =$ slope of line between $(x_1, y_1)$ and $(x_2', y_2')$
5. While ($s_1 * s_3 \neq -1$) do
   a. $(x_2', y_2') =$ next point on $l_2$
   b. $s_3 =$ slope of line between $(x_1, y_1)$ and $(x_1', y_1')$

The final line formed between $(x_1, y_1)$ and $(x_2', y_2')$ is $l_3$, the top boundary line which is perpendicular to both to $l_1$ and $l_2$. A similar algorithm is used to approximate top boundary line $l_3$ for counter clock-wise skewed document. After approximating left, right and top boundary line, next the transformation process is performed to correct the skew and the process is explained in next section.

2.2.3 Non-rotational transformation for skew correction

The transformation process collects all the points of the skewed document image 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 in repositioning the
skewed points into appropriate positions. For comprehension, a set of skewed lines are shown as skew corrected lines using the approach in figure(2.2.3a).

Lines $L_1...L_{i}...L_k$ are the possible lines between line1 and line2 which are parallel to line3. For each $L_j, j=1..k$, collect all points of the skewed document image on the line $L_j$ and place the set of collected points on a horizontal line to result the skew correction. This process of transformation from a single line suffers loss of points due to the limitation of discretization and line drawing algorithm[Don103] as shown in figure(2.2.3b). The loss of points is directly proportional to the skew and this reduces the readability by OCRs. In order to retain the readability in transformed text, additional points are generated by considering three lines instead of a single line.

Consideration of three lines increases the thickness of characters and results the transformed document image to appear slightly bold than the original. Figure(2.2.3c) illustrates, obtaining a transformation line from three lines.
A single transformation line is obtained by considering three lines. The first line is the actual line under consideration between \(i^{th}\) point on \(line\_1\) and \(j^{th}\) point on \(line\_2\). Second line is between the \((i-1)^{th}\) point on \(line\_1\) and \(j^{th}\) point on \(line\_2\). Third line is between the \(i^{th}\) point on \(line\_1\) and \((j-1)^{th}\) point on \(line\_2\). The skew corrected line of points is obtained by the union of these three set of points. Transformation lines are obtained for all possible lines in the document except for the first line. Algorithm for the transformation of clock-wise skewed document is given as follows.

**Input:** Skewed document area between \(line\_1\) and \(line\_2\) under \(line\_3\), \(P\) a blank document

**Output:** \(P\) skew corrected document

**Method:**

1. \(i^{th}\) point on \(line\_1\) = \((x1,y1)\)
2. \(j^{th}\) point on \(line\_2\) = \((x2',y2')\)
3. \(k = 1\)
4. While \((i\) is not end point of \(line\_1\) and \(j\) is not end point of \(line\_2\)) do
   a. \(S\) = set of points on line between \(i^{th}\) point of \(line\_1\) and \(j^{th}\) point of \(line\_2\) obtained by considering 3 lines
   b. Put \(S\) on line \(k\) in \(P\)
   c. Advance \(i\) to next point on \(line\_1\)
   d. Advance \(j\) to next point on \(line\_2\)
   e. \(k = k + 1\)
5. Final \(P\) is the skew corrected document

Similarly, the above algorithm is extended to transform the skewed points of counter clock-wise skewed document. The result of the implemented method for different
skewed input documents considering different languages is discussed in the next section.

2.2.4 Experimental Results

Experiments are conducted considering different skewed documents from different languages. Skew in the documents are obtained synthetically for experimentation. The performance of the proposed method is consistent for the skewed documents in the range of -30° to 30°. The method shows relatively good results in terms of retaining readability though the result suffers from information loss and distortions. Information loss is due to the limitations of descretization and line drawing algorithm. Distortions are due to generation of new points to reduce information loss. Figures illustrated from figure(2.2.4a) through figure(2.2.4h) show some samples of skewed print text document input images and the skew corrected output document images. Maximum size of document considered for experimentation is text in an A4 sized paper. But, for the sake of presenting the work smaller size documents are considered. These smaller sized input documents are paragraphs consisting of 15-20 lines from single column obtained from magazines and newspapers as documents of A4 size cannot have larger skew in the scanning device.

Figure(2.2.4a): Clock-wise skewed English document and corrected document
Statistical analysis from a corpus of Bangla text is useful and important in designing the OCR apart from other applications in Natural Language Processing and Cryptography. Since no such analysis is available in the literature, we collected two sets of data, the first set containing 100,000 words from popular Bangla books, newspapers and juvenile literatures and the second set containing 60,000 words from a Bangla dictionary.11 Each datum is a word converted into ASCII codes of Fig. 1. A typical example is A/NT//RGH/A,TMU,LK. Here, /NT/ denotes that N and T together make a compound character. The words in the data sets were stored in alphabetical order. For the first set of data each distinct word is stored with its frequency of occurrence.

Figure(2.2.4b): Counter clock-wise skewed English document and corrected document

Figure(2.2.4c): Clock-wise skewed Kannada document and corrected document

Figure(2.2.4d): Counter clock-wise skewed Tamil document and corrected document
Figure (2.2.4e): Counter clock-wise skewed Malayalam document and corrected document

Figure (2.2.4f): Clock-wise skewed Telugu document and corrected document
Figures (2.2.4a) through (2.2.4h) show the input skewed images and skew corrected images in 0° orientation. Similarly, the method can also correct the skewed images in other orientations. Figures (2.2.4i) and (2.2.4j) illustrate such skew corrections in orientation other than 0°. In case of input forms skew correction process is much easier since the size and the structure of the input forms are predefined. Drawing left boundary line, right boundary top boundary line is quite simpler with such known input forms. However the method explained in this section can be comfortably extended to correct the skew in input forms. Figure (2.2.4k) shows an instance of skew correction in an input form image.
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.
2.2.5 Readability Analysis of Skew Corrected Images

To analyze the result in terms readability, English documents at three different resolutions 200 dpi, 250 dpi and 300 dpi are considered. Document of each resolution is further considered with skew angles ranging from -30° to 30° at an interval of 5°. The required skew is obtained synthetically from a non skewed document. Result of each input document is attempted to read from a commercial OCR – “Readiris Pro10” and the readability efficiency is given in table-2.2.5a. The columns with heading ‘Specific case’ indicate the reading rate of skew corrected document images for the images obtained at different skew angles from a specific document. The columns with heading ‘Average’ indicates the average reading rate of skew corrected document images from different angle of skew for a set of 10 different document images.
Table-2.2.5a: Result of readability for skew corrected document at different resolutions

<table>
<thead>
<tr>
<th>Skew</th>
<th>200 dpi Specific case</th>
<th>200 dpi Average</th>
<th>250 dpi Specific case</th>
<th>250 dpi Average</th>
<th>300 dpi Specific case</th>
<th>300 dpi Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>-5</td>
<td>98.78</td>
<td>99.54</td>
<td>100</td>
<td>99.38</td>
<td>86.99</td>
<td>86.84</td>
</tr>
<tr>
<td>-10</td>
<td>98.78</td>
<td>99.56</td>
<td>99.59</td>
<td>99.47</td>
<td>82.52</td>
<td>82.46</td>
</tr>
<tr>
<td>-15</td>
<td>100</td>
<td>99.76</td>
<td>100</td>
<td>99.51</td>
<td>82.52</td>
<td>82.43</td>
</tr>
<tr>
<td>-20</td>
<td>99.59</td>
<td>99.47</td>
<td>97.96</td>
<td>99.44</td>
<td>81.70</td>
<td>81.67</td>
</tr>
<tr>
<td>-25</td>
<td>100</td>
<td>99.68</td>
<td>99.59</td>
<td>99.38</td>
<td>79.67</td>
<td>80.66</td>
</tr>
<tr>
<td>-30</td>
<td>99.59</td>
<td>99.52</td>
<td>98.78</td>
<td>99.46</td>
<td>88.61</td>
<td>88.55</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>99.68</td>
<td>99.59</td>
<td>99.48</td>
<td>86.99</td>
<td>86.66</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>99.58</td>
<td>99.59</td>
<td>99.42</td>
<td>85.77</td>
<td>86.44</td>
</tr>
<tr>
<td>15</td>
<td>100</td>
<td>99.76</td>
<td>99.59</td>
<td>99.36</td>
<td>86.17</td>
<td>86.03</td>
</tr>
<tr>
<td>20</td>
<td>99.59</td>
<td>99.52</td>
<td>99.18</td>
<td>99.33</td>
<td>81.70</td>
<td>82.36</td>
</tr>
<tr>
<td>25</td>
<td>100</td>
<td>99.54</td>
<td>100</td>
<td>99.34</td>
<td>79.26</td>
<td>80.45</td>
</tr>
<tr>
<td>30</td>
<td>100</td>
<td>99.48</td>
<td>100</td>
<td>99.34</td>
<td>79.26</td>
<td>80.45</td>
</tr>
</tbody>
</table>

99.69  99.58  99.52  99.41  83.60  83.82

It is evident from the table-2.2.5a that the readability reduces in skew corrected documents as the resolution of the input document decreases. To have better readability, the input documents should be obtained at 250 dpi or above. The reason for reduction in readability for lower resolution documents is due to the loss of points in the document during transformation. Though additional points are generated during transformation, all possible missing points are not generated and there is possibility of discontinuities in transformed characters resulting in misrecognition.

Consideration of three lines during transformation is to generate additional points in order to fill some points that are lost. This process keeps the loss of points with in a low range, other wise the loss of points range is very high and the readability is inversely proportional to the number of points lost. An analysis is performed to study the loss of points in case of single line transformation is illustrated in table-2.2.5b for the documents considered in table-2.2.5a at 250 dpi and it is evident from the analysis that loss of points is proportional to skew and readability is proportional to loss of points.
Table-2.2.5b: Loss and readability for single line and three lines transformations

<table>
<thead>
<tr>
<th>Skew in degrees</th>
<th>Single line transformation</th>
<th>Three lines transformation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Loss of points</td>
<td>Readability</td>
</tr>
<tr>
<td>-5</td>
<td>125</td>
<td>99.59%</td>
</tr>
<tr>
<td>-10</td>
<td>498</td>
<td>98.78%</td>
</tr>
<tr>
<td>-15</td>
<td>1204</td>
<td>97.96%</td>
</tr>
<tr>
<td>-20</td>
<td>2189</td>
<td>96.74%</td>
</tr>
<tr>
<td>-25</td>
<td>3354</td>
<td>96.34%</td>
</tr>
<tr>
<td>-30</td>
<td>4743</td>
<td>95.93%</td>
</tr>
</tbody>
</table>

In case of single line transformation the range of loss of points is 125 to 4743, whereas in case of three lines transformation the range of loss of points is 92 to 1347. Because of the small range of loss of points in case of three lines transformation, the approach yields a consistent readability for a document image obtained in a specific resolution. Comparison of readability between single line transformation and three lines transformation is shown in figure(2.2.5a).

Figure(2.2.5a): Readability comparison of single line and three lines transformation
The method is also tested with small icons or pictures along with text in the document in which the pictures does not lie out side the left boundary of text and a sample of such document is shown in figure(2.2.5b).

![Figure(2.2.5b): Skewed text document with picture and corrected document](image)

### 2.2.6. Conclusion

The proposed method for skew detection and correction of printed document images is relatively less expensive in computation. The developed method is capable of correcting the skew in images of printed documents of any script. The result indicates the method retains the readability in skew corrected document images above 99% for document images obtained at 250 dpi or above. Majority of skew detection and correction approaches noticed in literature require the accurate determination of the skew angle in the document image and then with the determined skew angle the skew of the document image is suggested for correction. The distinguishable feature of developed method is that, without determining the angle of skew a direct approach is made to correct the skew in document image. The method is capable of correcting skewed documents embedded with some figures also. The method has a limitation of correcting the skew between -30° and 30° and it is observed that most of skew due to mis-feed of document to the scanner ranges from -30° to +30°, and the proposed work is capable of accommodating range that is required. The method is sensitive to resolution of the document. The efficiency of the method drops as the resolution of the documents reduces below 250 dpi. An important requirement for the method is the complete document should be in the image.
All the stages in implementation viz., fixing left boundary, right boundary, top boundary and transformation are of computational complexity $O(n^2)$ and hence the total computational complexity of the method is $O(n^2)$. Any image processing method with time complexity $O(n^2)$ is considered as computationally less expensive and based on this, the method is claimed as less complex. The proposed method requires to be enhanced further to yield higher readability efficiency for document images obtained at less than 250 dpi. There is more scope to find a method to identify the missing points and regenerate these points for single line transformation.

The process of orientation detection and correction follows the skew correction process and the same is discussed in next section.

### 2.3 Orientation Detection and Correction

In the first level, the skew if any, in the document image is corrected with respect to any one of the four directions as explained in the section 2.2 and then the same is subjected for the detection of orientation. In case, the document image is not in proper orientation (portrait normal), then the method detects the direction of orientation and places the document image into proper direction. It is quite easy to detect the direction of input forms since the structures of input forms are predefined and detection orientation is possible through looking into defined features in the document images. But, detection of orientation is little difficult in general documents that does not have defined structure like headings, table etc. An attempt is made to detect and correct the orientations in such documents initially is explained in subsection 2.3.1 and then the methods used with respect to types of input forms are discussed in section 2.4.
2.3.1 Orientation detection and correction in English printed text document images

Considerable amount of research is reported in literature on the work of skew and orientation detection. The quantum of research work is quite high on skew detection, Jonathan and Taylor[Jon98] 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 $30^\circ$ and very few methods are limited upto $45^\circ$. This is because in normal situation the skew will not be more than $15^\circ$. Each of the methods discussed has its own limitations and advantages. Shivakumar[Shi105] 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.

Detection of page orientation would be simple task if the layout of the document is known aprori. 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[Adi102] 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[Le94] 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[Avi105] in 2005 suggested a fast method to detect skew and orientation in text images. The method is limited only to detect skew between $90^\circ$ to $-90^\circ$ to normal position but do not address the detection of flipped orientations. In 2000, Caprari[Cap100] had come out with a model for detection of page up/down orientation. This approach is limited to detect only orientations in $0^\circ$ and $180^\circ$.

\[2\]This part of work is communicated to Journal of Computer Science, ‘Detection and Correction Page Orientation in Monochrome Textual Images, Jan’2007.
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. Skew correction in a specific
direction is discussed in section 2.2. Orientation detection and correction is continued
in this section.

A simple X-cut and Y-cut technique\cite{Gon102; Jai98; Gor98} 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. A skew corrected document
image is shown in figure(2.3.1a). From the decision rule designed above, the method
predicts, the document is in landscape direction(90° or 270°).

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 figure(2.3.1a)
through figure(2.3.1e). A logical OR operation is performed on these five parts of the
document and the result of the operation is shown in figure(2.3.1f). 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.

Figure (2.3.1a): Skew corrected sample image

The pixel distribution graph for figure (2.3.1f) is shown in figure (2.3.1g). 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.
Let $d_{left}$ be the density of pixels in the region to the left of central region and $d_{right}$ 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.
\[ DL = \sum_{i=1}^{k} dleft_i \quad \text{and} \quad DR = \sum_{i=1}^{k} dright_i \]

Average \( DL = DL / k \) \quad \text{and} \quad \text{Average} \ DL = DR / k

For a document in portrait / landscape direction,

\begin{align*}
\text{If Average } DL & > \text{ Average } DR \text{ then} \\
\text{Document is oriented in } 0^\circ / 90^\circ \\
\text{Else} \\
\text{If Average } DL & < \text{ Average } DR \text{ then} \\
\text{Document is oriented in } 180^\circ / 270^\circ \\
\text{Else} \\
\text{Failure}
\end{align*}

The computations performed for detection of orientation is made on the logically ‘OR’ed image [figure(2.3.1g)] using point processing technique[Gon102; Jai98]. 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 figure(2.3.1h). Analysis of computational complexities of both the approaches is discussed below assuming the size of the divided images as \( n \times n \).

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 2.2.3 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.

### 2.3.2 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-2.3.2a gives the summary of results of skew correction process and table-2.3.2b gives the summary of the orientation correction process.

#### Table-2.3.2a: Results of skew correction process

<table>
<thead>
<tr>
<th>Skew direction</th>
<th>No. of samples</th>
<th>Result without error</th>
<th>Result with error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0^0$</td>
<td>154</td>
<td>139 (90.25%)</td>
<td>15 (09.75%)</td>
</tr>
<tr>
<td>$90^0$</td>
<td>157</td>
<td>145 (92.36%)</td>
<td>12 (07.64%)</td>
</tr>
<tr>
<td>$180^0$</td>
<td>158</td>
<td>142 (89.87%)</td>
<td>16 (10.13%)</td>
</tr>
<tr>
<td>$270^0$</td>
<td>155</td>
<td>140 (90.32%)</td>
<td>15 (09.68%)</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>624</strong></td>
<td><strong>566 (90.71%)</strong></td>
<td><strong>58 (09.29%)</strong></td>
</tr>
</tbody>
</table>

Table-2.3.2a indicates an average efficiency of 90.71% in skew correction stage without error. Around 9.29% is reported as skew correction with an error ranging $\pm 4^0$. 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-2.3.2b: Results of Orientation detection process

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

Table-2.3.2b 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° 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° and < 0.4° then there is a possibility in the method to misclassify the orientation. The skew correction process results with an error <0.2° 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.

2.3.3. 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(n²) as both the stages use point
processing technique[ Gon10; Jai98] assuming the size of the image as $n \times n$. 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.

2.4 Detection and correction of orientation in input form images

The structural layout of input forms is predefined. Detection of page orientation would be simple task if the layout of the document is known a priori. In such cases, the knowledge about the structure layout of input forms is exploited to extract certain features from the document and use the same for detection of orientation. The methods developed based on these feature extraction are very much specific to application domain and cannot be used for generic applications.

One such simple way of finding the direction of orientation as portrait or landscape is on the size feature of the input forms. From the figure(1.1.3a) in chapter 1, it is evident that the automatic data extraction system maintains a knowledgebase about the different types of input forms available within the system. Based on the size parameters available in the knowledgebase, the direction of the input forms can be decided as portrait or landscape. The exact direction of orientation is detected in the next level by looking into the features in the input forms. Every input form type is supposed to have some dominant features at certain positions. The positions of these features in document images that are in other orientations will be definitely at different locations and this feature is exploited for detection of orientation.

Few instances of input forms are considered for detection and correction of orientation. For the sake of comprehension a Bank challan shown in figure(2.4a) is considered initially to develop an application specific detection and correction of orientation.
The image of a Bank challan shown in figure(2.4a) is obtained at 300 dpi resolution and has the size 675 x 375 pixels. The input form shows a continuous line on the left side of the form approximately from the position 70 and runs up to position 300 in y-direction. This feature is not available on the other three sides of the input form. The size of the input form subjected for detection of orientation is compared with the size of input forms stored in knowledgebase for detection of the direction as portrait or landscape. The angle of orientation is decided based on the rules shown in table-2.4a. Once the angle of orientation is detected, the process of correction is same as explained section 2.3.

**Figure(2.4a): A sample input form**

<table>
<thead>
<tr>
<th>Direction</th>
<th>Place of line</th>
<th>Angle of orientation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Portrait</td>
<td>Left edge of image</td>
<td>0°</td>
</tr>
<tr>
<td></td>
<td>Right edge of image</td>
<td>180°</td>
</tr>
<tr>
<td>Landscape</td>
<td>Bottom edge of image</td>
<td>90°</td>
</tr>
<tr>
<td></td>
<td>Top edge of image</td>
<td>270°</td>
</tr>
</tbody>
</table>
Table-2.4a indicates that if the orientation is portrait then the direction is identified through looking the line on the left side for 0° orientation or on right side for 180° orientation. If the orientation is landscape, then the direction is identified through looking the line on bottom side for 90° orientation or on top side for 270° orientation.

Another instance of input form image for orientation detection is shown in figure(2.4.b) scanned at 300dpi resolution. From the apriori knowledge of the form it is learnt that a logo of the organization is available in the region [(110,80), (240,200)] from the left top corner. The size of the input form image is 3250 x 2275 pixels. As usual the size parameters are used to decide the direction of image as portrait or landscape as in earlier case. Then a search is made to locate the position of the logo on the input form image subjected for detection of angle of orientation. The search is a simple computation of density of black pixels in the specified area. It is learnt that the area of logo consists of 15600 pixels and the logo in the document consists of more than 14000 black pixels. The black pixel density in the logo area is fixed for a threshold of 14000. During the search, the number of black pixels is counted from the specified area. If the density value exceeds the threshold then it is decided that the logo is available in that position otherwise not. Based on the search result, the angle of orientation is detected and the decision on search is shown in table 2.4b.

Since it is quite impossible to build generic orientation detection model, specific simple and straightforward methods can be implemented with respect to each type input forms based on the knowledge of structural layout of the input form documents as indicated in the above instances. The experimental study shows 100% orientation detection when known types of input forms are considered. But the result is around 95% with inclusion of unknown types of input forms showing 3% of rejection and 2% of misclassification.
Visveswariah Technological University, Belgum
Practical Examination Marks Sheet for

<table>
<thead>
<tr>
<th>Batch Id</th>
<th>Date: 04/04/06</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time: 09:00</td>
<td></td>
</tr>
<tr>
<td>Region: NA</td>
<td></td>
</tr>
<tr>
<td>College: NA</td>
<td></td>
</tr>
<tr>
<td>Subject: NA</td>
<td></td>
</tr>
</tbody>
</table>

**Marks**

<table>
<thead>
<tr>
<th>Seat number</th>
<th>A</th>
<th>C</th>
<th>Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Examiners**

Name: | Signature: | Institution: |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Instructions for marking entry**
- The candidate should mark the circle under the column 'A' only.
- To cancel an entry, darken the circle under the column 'C' only.
- Sample should read:
  - 100 to be entered as: 100 1 0 0
  - 0 to be entered as: 0 0 0 0
  - 90° to be entered as: 0 9 0 0

**Figure (2.4b): Another sample of input form**

**Table 2.4b: Decision for angle of orientation**

<table>
<thead>
<tr>
<th>Direction</th>
<th>Position of logo</th>
<th>Angle of orientation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Portrait</td>
<td>(110,80), (240,200)</td>
<td>0°</td>
</tr>
<tr>
<td></td>
<td>Otherwise</td>
<td>180°</td>
</tr>
<tr>
<td>Landscape</td>
<td>(2075,80), (2195,200)</td>
<td>90°</td>
</tr>
<tr>
<td></td>
<td>Otherwise</td>
<td>270°</td>
</tr>
</tbody>
</table>
2.5 Chapter Summary

In this chapter, how document images are prepared for data extraction is explained at the document image level. A novel language independent non-rational approach is developed to detect and correct printed text document images. A method for orientation detection and correction is developed with the support of skew correction is also presented for printed English text documents. The method for skew correction stage can be extended for input forms but, orientation detection method cannot be extended directly for input forms. Since the structural layout is predefined for input forms and the types of input forms are also finite in a specific application, specific methods can be easily developed for preparing the input form document images for automatic data extraction. Two such instances with respect to input forms are explained for understandability and similarly the methods can be implemented for other types of input forms.