CHAPTER 2

LITERATURE SURVEY

2.1 INTRODUCTION

Information Retrieval from Text document images could be successfully performed only after understanding the text present in the document image. In general, physical structures of the image, functional relationships between the blocks and the underlying script must be analyzed in prior before understanding the text. Therefore, for an input document image, layouts must be analyzed to identify the textual blocks. After layout analysis, scripts are identified from the text areas, and suitable text understanding engines have been employed to understand the text, which enables information retrieval. So, from this perspective, Section 2.2 discusses the various physical layout analysis techniques, Section 2.3 discusses the related and previous works of Script Identification and Section 2.4 describes the various text understanding and retrieval methods from document images. In this thesis, a new classification scheme for Script Recognition and Text Understanding techniques has been introduced.

2.2 PHYSICAL LAYOUT ANALYSIS

Document image layout analysis (Physical layout analysis) is a crucial step required in document image analysis for understanding the text. Layout analysis is the process of identifying layout structures by analyzing page images and could be physical (text, graphics, pictures) or logical (title, abstract, author etc).
Approaches for Physical layout analysis could be of three types: (1) top down methods, (2) bottom up methods and (3) integration methods. Top-down method uses multi-level X-Y projections (Tang et al 1994) and divides the whole page to smaller regions i.e. a page is segmented from larger components to smaller sub components. Basically, in bottom-up analysis, connected components are extracted from the image and are subsequently aggregated in higher level structures, creating words, text lines, paragraphs, and so on. Bottom approaches appear to be more appropriate for non Manhattan layouts whereas top down approaches can process only Manhattan layouts.

2.2.1 Top Down Approaches

Top-down methods are generally faster and have the advantage of directly extracting a geometric hierarchy among regions that is frequently tightly linked with the logical structure of the document (Cesariani et al 2002). Top down approaches traverses the document in horizontal and vertical direction of the image to analyze the regions without much complication. Top down approach involves X-Y projection profiles for layout analysis.

2.2.1.1 X-Y Projection Profiles

The X-Y tree (Nagy and Seth 1984) is a well-known top-down method for page layout analysis. The basic assumption behind this approach is the fact that structured elements of the page are generally laid out in rectangular blocks. The document is split into successively smaller rectangular blocks by alternately making horizontal and vertical “cuts” along white spaces. The spaces are found by using a thresholded projection profile (a projection profile is the histogram of the number of black pixels along parallel lines through the document). The result of such segmentation can be
represented in a X-Y tree, where the root corresponds to the whole page, the leaves are for blocks of the page, whereas each level alternatively represents the results of horizontal or vertical segmentation. This requires a number of thresholds in both the directions to identify the layouts. Later thresholds in reducing the risk of bad cuttings (Cesarini et al 1999) have been eliminated by calculating the threshold on the basis of the average height and width of characters in the page to be processed.

This approach obtains good results for a broad range of documents when the threshold setting has not been changed and provides worst segmentation for the documents containing abrupt discontinuities and dashed lines in background. As an extension, a modified X-Y tree (MXY tree) has been proposed by Cesarini in which the main features of M-XY trees (Cesarini et al 2002) act as storage of cutting lines as leaves of the tree and the description of inter-leaves adjacency relations by means of appropriate links.

Later, Marinai introduced a layout based retrieval system (Marinai et al 2004) using MXY trees. Here, each document image has been converted into feature vectors describing the global features of the page and MXY tree structure and feature vectors are stored in database (Marinai et al 2004, Marinai et al 2005a). During retrieval, a query by example approach is considered. User selects one sample page by browsing the collection and the query feature vector has been compared with the feature vectors in the database and the retrieved documents are shown to the user.

Even though, top down approaches could work fastly, two main limitations of top down algorithms are, (1) Algorithm requires a well aligned image, (2) Document could be segmented only along the separators covering the whole width (or height) of the image corresponding to the node to be split (not suitable for non Manhattan layouts.) Therefore, in order to overcome
these shortcomings, bottom up approaches are applied for physical layout analysis.

2.2.2 Bottom Up Approaches

Bottom up approaches have the capability to process non-Manhattan layouts containing larger headlines, wider word gaps and text blocks being intermingled with pictures. Further, these layouts cannot be processed through proper vertical and horizontal segmentation. There are six representative algorithms (Shafait et al 2006) or widely used algorithms for Page Segmentation using bottom up approach. They are Connected Component analysis (CCA), Run length Smearing algorithm, Docstrum, Voronoi algorithm, Constrained Text line finding and White space Analysis.

2.2.2.1 Connected Component Analysis

Connected component analysis starts the analysis pixel by pixel, characters are combined into words, lines, paragraphs and finally blocks or regions. CCA can analyze most of the text regions but this is a much time consuming process.

2.2.2.2 Run Length Smearing Algorithm (RLSA)

The Run-Length Smearing Algorithm (RLSA) (Wong et al 1982) works on binary images, where white pixels are represented by 0s and black pixels by 1s. The algorithm transforms a binary sequence of white pixels between the black pixels into black, if the number of adjacent white pixels is less than or equal to a predefined threshold. These steps have the effect of linking together neighboring black areas into more black. The RLSA is applied row-wise to the document using a threshold and column-wise using another threshold, yielding two distinct bitmaps. These two bitmaps are
combined in a logical AND operation. Additional horizontal smearing is done to obtain a smoothed final bitmap using a smaller threshold. Then, connected component analysis is performed on this bitmap to obtain document zones. Since Run length algorithm involves more computations, Docstrum algorithm has been proposed by Gormann for page segmentation.

### 2.2.2.3 Docstrum Algorithm

The docstrum algorithm proposed by O’Gorman (Gorman 1993) is a bottom-up approach based on the nearest neighborhood clustering of connected components extracted from the document image. In document images, after noise removal, connected components are identified and separated into two groups, one with characters of the dominant font size and another one with characters in titles and section headings, using a character size ratio factor. Nearest neighbors are found for each connected component in text lines and merged. Finally, text-lines are merged to form text blocks using a parallel distance threshold and a perpendicular distance threshold. But this algorithm would fail if variations occur in font faces, styles and sizes. But the Voronoi algorithm introduced by Kise could adapt variations to some extent.

### 2.2.2.4 Voronoi Algorithm

The Voronoi diagram-based segmentation algorithm by Kise (Kise et al 1998) is a bottom-up algorithm for page segmentation. CCA has been performed in the document image to identify connected components of the document image. In the first step, it extracts sample points from the boundaries of the connected components using a sampling rate. Later, a Voronoi diagram has been generated using sample points obtained from the borders of the connected components. The Voronoi edges that pass through a connected component are deleted based on the threshold setting to obtain an
area Voronoi diagram. Finally, Voronoi edges from the area Voronoi diagram are deleted (based on threshold) to obtain boundaries of document components. Even though Voronoi diagram identifies layouts efficiently, it heavily relies on threshold and parameters for identifying Voronoi edges. Based on these shortcomings, Bruel introduced a hybrid approach called as White space analysis technique which does not rely on threshold and can work across text variations.

### 2.2.2.5 White Space Analysis

A promising hybrid approach describes White space cover analysis (Baird 1990, Baird 1994), which analyzes the white background into a set of white rectangles (called covers) whose union cover it completely. Breuel’s algorithm finds the maximal empty white space (Breuel 2002, Breuel 2003) rectangles which lies between the black rectangles as a first step by sorting the black rectangles. (Black rectangles in the image are identified through connected component analysis performed over the image). In the second step, the white rectangular covers are combined (merged) one by one to generate a corresponding sequence of segmentations. Segmentation is the uncovered area left by the union of the covers combined so far and treated as text block. Before a cover is unified to the segmentation, a trimming rule is applied to avoid early segmentation of narrow blocks. The unification of covers continues until a stopping threshold gets satisfied.

At the final segmentation, connected components within the remaining uncovered parts are candidate text regions. Subsequently, layout analysis approach by Bruel finds text-lines in the text blocks (Bruel 2002) through the bounding box of all the characters in text lines through constraint text line finding algorithm. But the white space analysis algorithm is sensitive to heuristics which gets reflected in the performance of page segmentation.
Therefore, to conclude with, for clean documents with regular arrangements, X-Y cut algorithm may be a good choice and fast to implement but this would fail for noisy and non Manhattan layouts. Bottom up approaches adapts non Manhattan layouts and noisy images but computational complexity is very high since this relies on various assumptions, thresholds and parameters. Based on these problems, a hybrid approach (called as Rectangular white space analysis) has been proposed in this thesis for layout analysis which adapts heterogeneous kinds of document images with reduced pixel visits. This also eliminates the assumption of heuristics, threshold and adapts variations in font faces and sizes.

2.3 SCRIPT IDENTIFICATION

Script Identification is a key part of the automatic processing of the document images in an international environment, when the document image is multi-script or multi-lingual. A document’s script must be known in order to choose an appropriate text understanding system (Hochberg et al 1997).

Script Identification approaches are twofold: namely, local and global approaches (Joshi et al 2006). In local approaches, connected components (Line, word and char) are analyzed in the document images, to identify the script. The components are available only after line, word and character segmentation of the underlying document image. Here, success of Script classification mainly depends on the character segmentation or connected component analysis (Maximal region of connected pixels). In contrast, global approaches analyze the block of text without finer segmentation. Global approaches works well on blocks of text by employing Texture based classification system.
2.3.1 Global Approaches (Texture Based Classification)

Texture is defined as the repetition of a pattern or patterns over a region. Texture is a complex visual pattern composed of sub patterns or Textons. The sub patterns give rise to the perceived lightness, uniformity, (Joshi et al 2006) density, roughness, fineness, smoothness, granulation etc as the texture as a whole. Texture of an image is also defined as a function of the spatial variation in pixel intensities (gray level values). The notion of texture appears to depend upon three ingredients: (i) some local ‘order’ is repeated over a region which is large in comparison to the order’s size, (ii) the order consists in the nonrandom arrangement of elementary parts, and (iii) the parts are roughly uniform entities having approximately the same dimensions everywhere within the textured region.

Features used for textural analysis may vary depending upon the application. Frequently used features would be Statistical features (co occurrence matrices like entropy, contrast and correlation) and Frequency domain (energy bands using Gabor filters), which measures the periodicity of the image. Texture analysis appears to be good for the problem of script identification since the texture analyses the global characteristics rather than the individual characters. Script/ language of the document images can be clearly identified using textures. Textures over a uniform block of text (where the line and word spacing are normalized) possess different textures for different scripts since it produces variations in character density and stroke orientation.

Script Identification can be performed at the levels of block (script of the document/ paragraph), line (script of a line) or word (script of a particular word). From this viewpoint, methods reported in the survey for texture measurement are classified as Grey level Co-Occurrence Matrices (GLCM), Multichannel Gabor filters, Rotation invariant texture features,
Wavelet based coefficients, Multichannel Log Gabor filters and Multiclass Gabor filters as indicated in Figure 2.1. In global approaches, block level script classification has been done with four techniques namely GLCM, Multichannel Gabor filtering, Wavelet Transforms and Multi Channel Log Gabor filters as shown in Figure 2.1.

![Figure 2.1 Global approaches for Script Identification](image)

### 2.3.1.1 GLCM and Multichannel Gabor Filters

Initially an attempt towards texture classification for script identification from Asian (Chinese, Japanese and Korean) and European (Roman) scripts without finer segmentation (Peake and Tan 1997) has been made by measuring two different texture features: Gabor filtering and Grey level co-occurrence matrices (GLCM’s) from uniform block of text. Uniform
block of text in an image (line and word spacing are normalized) has been obtained through preprocessing and segmentation of documents into paragraphs, over which texture analysis methods are applied.

Gabor filters (Dhanya et al 2002) acts as a good model for Texture segmentation and Texture classification. Here Multichannel Gabor filter has been employed which requires an N×N pixel as input image, (where N is a power of 2), an angle theta and frequencies at 4, 8, 16, and 32 cycles/ degree. This result in 16 output images (4 from each frequency) from which the texture features were extracted by calculating the mean and standard deviation of each output image which yields 32 features per input image. Testing was conducted using all 32 features and various subsets of the features (16 mean features and 8 features from a single frequency) for script identification.

GLCM’s produces a set of matrices with each matrix as N* N size, where N is the number of gray levels in the image, for five distances (d=1..5) and four directions for 0, 45, 90 and 135 degrees respectively. Testing was conducted using different combination of the features in both the texture measurement and it was observed by them that the Gabor filters produces higher accuracy of 95% than the GLCM. This accuracy has been obtained for a single font and the accuracy falls to 72% when multiple fonts are incorporated.

Later, Rotation invariant texture features (Tan 1998) were computed based on an extension of the popular multi-channel Gabor filtering technique to identify six languages. Six languages (Chinese, English, Greek, Russian, Persian, and Malayalam) were chosen to demonstrate the potential of such a texture-based approach in script identification.

Consequently, the above multichannel Gabor filtering was extended to Indian languages with six degrees of frequency (Chaudhury and Sheth
1999) to identify the script between English, Hindi, Telugu and Malayalam at paragraph level. It was seen that English and Malayalam respond to zero degrees of Gabor filters, English produces peak in 90 degree filters due to the dominance of vertical strokes, Telugu scripts which were mostly circular have nearly equal response to all the Gabor filter directions.

Similarly, Chan and Sivaswamy (1999) and Joshi et al (2006) have used Gabor features for classification of most of the Indian script documents. They assumed a block of text to consist of characters from the same script and employ a multi-channel Gabor filter bank to discriminate between the various scripts. Manthalkar and Biswas (2002) have used rotation-invariant texture features, using Gabor filter bank, to differentiate between text blocks for various Indian scripts.

Therefore, all the multichannel Gabor filtering methods reported above requires a uniform block of text (size of the image, line spacing, word spacing, style etc.) for texture measurement. Accuracy of script identifier fails in non uniform blocks of text. In order to have a script identifier which could adapt the non uniformity in text blocks, an additional feature called Wavelet transform based texture features has been introduced in addition to Gabor filters.

2.3.1.2 Wavelet Transforms

Later, Wavelet Energy Features, Wavelet Log Mean Deviation Features, Wavelet Co-Occurrence Signatures and Wavelet Scale Co-Occurrence Signatures (Busch et al 2005) were included as textual features in addition to the GLCM and multichannel Gabor energy filters to identify the script/languages between English, Chinese, Greek, Cyrillic, Hebrew, Hindi and Japanese over the non uniform block of text which have been normalized. In this method, after binarization, deskewing and
segmentation, a number of preprocessing operations were performed over the text to produce a uniform appearance. To normalize the text, each line is scaled to a standard height, spacing between the characters and words were normalized and short lines are padded with extra lines to the width of the longest line. Six texture features defined here were used to measure the texture from the normalized block of text.

Texture features reported above works well to identify the scripts but fails when it was used to handle large number of scripts. Moreover, normalization is required to make the text uniform. Efficiency of Gabor filters have been explored in detail to adapt large number of scripts using Multichannel Log Gabor filters.

2.3.1.3 Multichannel Log Gabor Filters

India has 18 official languages, which arise from ten scripts. A Script identifier to discriminate all the 10 Indian scripts (Joshi et al 2006) has been designed using the oriented energy distribution. (Log Gabor filters). The oriented energy distribution characterizes the texture of the script as it indicates the dominance of individual sub patterns. This energy distribution has been calculated for text blocks using log Gabor filters designed at 8 equi-spaced orientations with 0, 22.5, 45, 77.5, 90, 112.5, 135.5, 180 degrees respectively. Several features such as Statistical features, Local features, Horizontal Profiles are extracted from the oriented energy distribution. In the highest level, gross information was used for a broad categorization, whereas in the lower levels, categorization was performed using finer analysis of the underlying script.

All the methods which were stated above operates on the block level of text to identify the scripts whereas some other texture features such as Multichannel Gabor filters and Discrete Cosine Transform combined with
Gabor filters have also been proposed by the researchers to identify the script at word level. Word level script identification allows the document to contain more than one script and the script of each and every word has been identified.

### 2.3.1.4 Multi Channel Gabor Filters

An initial attempt in word level script identification has been made to separate Tamil and English scripts (Dhanya et al 2002) using the directional features (multi channel Gabor filter responses) of the words. Here the Gabor filter was designed with various orientations (0, 30, 60, 90, 120, 150 degrees) and various frequencies (f =0.25 and 0.5 cpi). It was observed that the Tamil script has more horizontal lines and strokes which corresponds to 0 degree filters and English has more vertical and slant strokes which corresponds to 90 degree filters. Also this method, works under the assumption that the word contains a minimum number of four connected components. This also assumes that the word was free from numerals.

Above stated multichannel Gabor filters (Pati et al 2004, Pati and Ramakrishnan 2006) was extended with four frequencies and four orientations (0, 45, 90, 135 degrees), which yields 32 filters (16 odd filters and 16 even filters) producing a 32 dimensional feature vector, was applied over the word images in bilingual document images which consists of English and an Indian script (Hindi, Tamil or Odiya). Later this 32 dimensional vector was analyzed with a Linear discriminant classifier to discriminate the script of word images.

Multichannel Gabor filtering designed with four frequencies and four orientations was also applied over the bilingual document images (Ma and Doermann 2003) Arabic-English, Chinese-English, Hindi-English and Korean-English bilingual dictionaries to identify the script at word level.
However all the methods require minimum number of characters in a word to identify the script.

Therefore, global approaches in script classification are relatively fast and reduce the cost of document handling. This does not require character segmentation or connected component analysis and even the presence of foreign characters in the document does not affect the overall texture of the block extracted. Moreover, there is no need to apply different processing methods based on the script of the document to produce better results. Even though, global approaches works well based on texture measurement, this relies on uniform block of text (Busch et al 2005) and extensive preprocessing (to make the text block uniform) is required to measure the texture. The classifier would fail if variable spacing and styles occurs between the lines, words and characters since the texture varies. Performance of the classifier also relies on the quality of the document images. Moreover, designing of multiple channel filters is a complicated process and it could not be extended to other scripts easily.

Although, local approaches rely on the accuracy of character segmentation or connected component analysis, this could work well on the documents across variations and qualities.

2.3.2 Local Approaches (Statistics Based Classification)

Local approaches do a connected component analysis to identify characters, words and lines in a document and record the position (location), bounding box and list of black pixel runs as features to identify the script. This kind of analysis is insensitive to noise and low quality document images.

Local approaches also perform Script Identification at the levels of block, line (script of a line) and word (script of a particular word). From this
viewpoint, methods of local approaches are classified as Upward Concavities, Token based approach in block level and Projection Profiles, stroke based, Structural based and Water Reservoir methods in line based script identification. Word based script identification mainly relies on Morphological Reconstruction whereas Prototype classification is adopted for Character level script identification as indicated in Figure 2.2. Initially, block level script identification has been attempted and it tries to identify the script of the given document in a mixture of various script documents. Various methods which lie under the local approaches of script identification are detailed in this section.

Figure 2.2 Local approaches for Script Identification

2.3.2.1 Upward Concavities

In script identification, an initial attempt was made (Spitz 1994) to distinguish Asian (Chinese, Japanese and Korean) and European (Roman) scripts in a mixture of document images by examining the presence and
location of upward concavities of characters in the image of text. Upward concavities were formed for a character, "When two runs of black pixels appear on a single scan line of the raster image and there is a run on the line below which spans the distance between these two runs". The position of each upward concavity was related by its vertical distance. Spitz observed that European and Asian scripts have very different distributions of upward concavities. In European scripts, there were usually at most 2 or 3 upward concavities in one character, and Asian scripts, on the other hand, have on the whole many more upward concavities per character (with no consistent arrangement) due to the higher level of character complexity.

Further distinctions among the Asian scripts were performed based on the analysis of the distribution of optical density of the text image (Spitz 1997). Initially, the given document image was discriminated as Han (Asian) or Latin (Roman) script based on the spatial relationship of features related to the upward concavities in character structures. 23 different Latin based languages were classified based on character shape codes.

Language Identification within the Han script was performed based on the text line, by accounting the number of black pixels obtained by summing the lengths of the runs that comprise all of the connected components within the cell. Later, upward concavity was utilized to distinguish three scripts such as Chinese, Latin and Tamil (Tan et al 1999). The distributions of upward concavities differ among these three scripts. For Chinese script, the distribution was more spread out, while the distribution for the Latin script was concentrated at the baseline and the x-height of each line. Distribution of upward concavities for Tamil script tends to be concentrated in the middle of the line. Analysis of upward concavities was most effective in identifying Chinese scripts, but less effective in distinguishing Latin and Tamil scripts.
Even though, upward concavity works well over the discrimination of Asian and European scripts (Spitz 1997), it relies on a uniform font size over the text area to be investigated. Lot of preprocessing is required before feature analysis. Moreover, this relies on a specific, well-defined pixel structures as features (Positions, frequency, density) for script identification. Therefore, in order to have a script identifier which learns the distinctions automatically among an arbitrary number of scripts, token based approach has been proposed.

2.3.2.2 Token Based Approach

A script identifier which automatically learns through tokens (templates), to distinguish thirteen scripts namely Arabic, Armenian, Burmese, Chinese, Cyrillic, Devanagari, Ethiopic, Greek, Hebrew, Japanese, Korean, Roman and Thai has been introduced (Hochberg et al 1995, Hochberg et al 1997). This method developed a set of representative symbols (templates) for each script by clustering textual symbols of that script, taken from a set of training documents and represented each cluster by its centroid. Textual symbols include discrete characters as well as adjoined characters and character fragments of the script and whole words in connected scripts such as Arabic. To identify a new document's script, the system compared a subset of symbols from the new document to each script’s templates and chooses the script which provides the best match. This system was unable to generalize across major differences in font types.

Further, this method has been extended to identify the various languages belonging to that script through feature analysis. Five features such as Relative vertical centroid, Relative horizontal centroid, number of white holes, number of black pixels in a unit area and aspect ratio (width/height) have been computed for each component (character). Finally, mean, standard deviation and skew of these five features for all the components have been
computed to produce a 15 dimensional vector for one textual entity and a classifier distinguishes the language based on these vectors.

Methods which were reported above (upward concavity, token matching) relies heavily on connected component analysis or the segmentation of characters and words in a document to identify its underlying script. Projection profile was used to discriminate the scripts at line level in very few languages.

In line based script identification, a document image can contain more than one script but it requires the same script on a single line. Projection profiles, topological and stroke based features, shape based characteristics, and water reservoir methods have been used to identify the script at line level. Many works related to this kind of script identification has been performed only in Indian scripts.

2.3.2.3 Projection Profiles

Horizontal projection was attempted (Elgammal and Ismail 2001) to separate two languages English and Arabic at text line level. Here, the horizontal projection profiles of Arabic text have a single peak corresponding to the baseline of the Arabic writing, where characters are connected together. In contrast, projections of English text have two major peaks corresponding to x-line and baseline. The projections of Arabic text lines were smooth while the projections of English text line have sharp jumps.

An advantage of using the projection profiles states that it was insensitive to touching or broken characters, and independent of the page orientation where the number of peaks and the density do not change even when the document was upside-down. Profile analysis would not be helpful to discriminate the Indian scripts due to its complex nature. In order to explore
the Indian scripts, topological, stroke based and structural features of the
script have been analyzed.

2.3.2.4 Topological and Stroke based Features

In the context of Indian language document analysis, major
literature was due to Chaudhury and Pal (Pal and Chaudhury 1997, Pal and
Chaudhury 1999, Chaudhury and Sheth 1999). Initially, in Indian scripts, an
attempt was made to discriminate the scripts (Pal and Chaudhury 1997)
among Roman, Bangla and Devnagari at text line level. Many characters of
both Bangla and Devnagari have a horizontal line at the upper part. In Bangla,
this line is called Matra and in Hindi it is called shirorekha and also it is
called as headline. Headline is the feature used to isolate Roman lines from
the Bangla and Devnagari lines.

Separation of Bangla and Devnagari script was trickier due to their
structural similarity and character level features have been employed here for
discrimination. Characters were segmented only at the middle zone of the
word and six kinds of Principal stroke features have been considered for script
classification.

However, principal stroke features could be used to discriminate
only the Bangla and Devnagari script and this method could not be applied for
other Indian scripts. Consequently, as similar to stroke feature concept,
structural features of various scripts have been explored to classify twelve
Indian scripts.

2.3.2.5 Structural Features

India has 18 official languages. Out of 18 scripts, twelve Indian
scripts have been explored to develop an automatic script recognizer at text
line level (Pal and Chaudhury 1999) using Structural features. This method assumes that the document may be printed in three language forms: English, Devnagari and one of the other Indian official languages. Script recognizer has been designed to classify using the characteristics and shape based features of the script.

Ten triplets were framed by joining English, Devnagari with any one of the official Indian script. Based on script shape characteristics, ten triplets were arranged in five groups. Along with English and Devanagari scripts, each group consists of doublet script such as Punjabi, Bangla; Urdu, Kashmiri; Gujarathi, Odiya; Telugu, Kannada and Tamil, Malayalam. This uses structural features such as Projection profiles, Vertical lines, Run length, Headlines etc. The overall accuracy of the system was about 98.5% and these techniques were insensitive to font faces and sizes. Further this structural features (i.e.) shape based feature concept has been extended with Water Reservoir concept to accommodate more scripts rather than triplets (three scripts).

2.3.2.6 Water Reservoirs

Later, Water Reservoirs combined with shape based and statistical features have been employed for the identification of script from printed document images containing five most popular scripts namely, Roman, Chinese, Arabic, Devnagari and Bangla (Pal and Chaudhury 2001). The concept of character water overflow may be explained by the analogy of water overflow from a reservoir. If we pour water from the top, water will be stored in the concavity of the character, which has been imagined as a reservoir. If we pour sufficient water, the water will flow out of the reservoir. Different features (like height of the reservoir, water overflow level etc.) of the reservoir were used in the separation scheme.
This script identifier has been further extended to accommodate different 12 Indian scripts (Pal et al 2003) in the same document instead of assuming the document to contain three scripts (triplets). Here various Structural features, Horizontal Projection profiles, Water Reservoirs (Top, Bottom, Left and Right), Contour Tracing (Left and Right profiles) were employed as features with a Decision Tree Classifier for script identification.

Thus, all the reported studies, accomplish script recognition either at the line level or at the paragraph level and no work has been successfully attempted to identify the script of a given word. Even though, stroke based features, structural features and water reservoirs act on multiple scripts and achieve good recognition accuracy, they were obtained only at line level. They cannot discriminate the scripts which have been mixed among words. In order to overcome this problem, word level script identification has been proposed.

2.3.2.7 Spatial Spread

Initially, word based script recognizer has been designed for Tamil and English language (Dhanya et al 2002) using Spatial spread features. Spatial spread analyses the pixel concentration of ascenders and descenders and middle zone in the word and concludes that the Tamil words have more pixel concentration in the lower zone than English words. In addition to that, the number of characters present per unit area in Tamil words was generally less than that in English words. This work achieved accuracy around 80%.

In order to have a script recognize specific to English, Kannada and Hindi scripts at word level, a neural network based script identifier (Basavaraj Patil and Subbareddy 2002) was introduced. Even though, this script recognizer works well, its performance relies heavily on trained font sizes and
styles. In order to adapt the font variations, Morphological reconstruction operation has been introduced.

2.3.2.8 Morphological Reconstruction

Later, this script recognizer has been extended to four scripts/languages (Kannada, Hindi, English and Urdu) with different font sizes and styles by relaxing their constraints over different font sizes (Dhandhra et al 2006a) using morphological reconstruction. Here, feature vectors were obtained from the words by extracting the characters containing strokes in four directions (Horizontal, Vertical, right diagonal and left diagonal) by erosion operation with the line structuring element. Later, these feature vectors were used to measure the similarity between the test and training vectors.

However the technique reported above could classify the scripts at word level based on alphabet recognition, but it fails if English numerals occur as a word (Numeral Identification). In order to classify the English numerals at word level, an automatic technique for script identification was proposed (Dhandhra et al 2006b, Dhandra et al 2006c), based on morphological reconstruction combined with directional stroke features, aspect ratio and eccentricity for printed bilingual documents of Kannada and Devnagari containing English numerals (printed and handwritten).

Later, this method has been improved with seven feature vectors (eccentricity, aspect ratio, strokes (horizontal, vertical, left diagonal, right diagonal) to classify the three bilingual document images (Dhandhra et al 2007, Dhandhra and Mallikarjun 2007) representing Kannada, Tamil and Devnagari containing English Numerals at word level based on the observation that every text has distinct visual appearance.
All the above reported methods contribute to the script identification at word level and very little work has been attempted to identify the script at character level.

### 2.3.2.9 Prototype Classification

In character level identification, script recognizer attempts to identify the language/script type of individual characters, without analyzing the coarse entities to which they belong. An attempt in this regard was made to classify the Japanese, Chinese and English characters using Prototype classification method (Liu et al 2005). This method consists of a training phase and testing phase. In training phase, a learning algorithm was used to construct prototypes from training samples (characters from each language) and clusters are formed. In testing phase, given sample was measured against cluster centre, to identify the script/language.

Script Recognition is a process that has no fixed solution (Tan et al 1999). The challenges, researchers face lies in finding the attributes that can clearly distinguish the scripts that are trying to recognize. However, no mathematical approaches or scientific techniques can be used to find these attributes. Research is mostly done on a trial and error basis. In addition, since different scripts have different attributes and style, the document attribute that is successful in identifying one script may not be successful in another script. In addition, the attributes must also be independent of the quality and variations of the document, to handle multiple fonts, sizes and poor quality.

However, script classification techniques reported in the literature were script dependent and could not be applied to discriminate Tamil from English scripts (which is discussed in this thesis) as such, since the nature of Tamil script varies from other Indian scripts. Moreover, script recognizers
introduced so far to discriminate Tamil and English scripts were not so much effective due to their constraints as stated in the following section.

### 2.3.3 Previous Works in Script Recognition

Script identification in monolingual document images (Tan et al 1999) consisting of English, Chinese and Tamil scripts were attempted using upward concavities and bounding box elongation. Upward concavities appear to be effective to discriminate the Chinese script whereas it was less effective between Roman (English) and Tamil since most of the distribution lies in the middle zone. Bounding box elongation (Aspect Ratio) between Tamil and English words (Tan et al 1999) would fail if both English and Tamil words comes in the same spatial distribution or if the word contains few characters. Word level script identification between English and Tamil Scripts using projection profiles (Dhanya et al 2002), works well for descender dominant Tamil characters. This fails for the descender dominant characters in English. Moreover, this approach produces an average performance of 75% because of lack of discrimination between scripts. Even though, discrimination of Tamil and English scripts (Pal and Chaudhury 1999) using ascenders and horizontal black runs obtained a recognition accuracy of more than 90%, this was restricted to line level script identification and cannot be applied to identify the script at word or character level. In the case of Water Reservoirs (Pal et al 2003), the number of left reservoirs has been considered as more for Tamil images than for English and this was also restricted to line level.

Therefore, in this thesis, the above reported shortcomings motivated us to the idea of developing Spatial Features Based Script Recognizer (SFBSR), to identify the scripts Tamil and English at character level. The idea is based on the assumption that the script of the character can be identified by analyzing the zonal density of the character cell, which could work across various fonts and sizes.
2.4 TEXT IMAGE UNDERSTANDING

Text Image Understanding could be performed in either of the two ways: (1) OCR and (2) Keyword Spotting technique. Initially, OCR was utilized to understand the text and text retrieval techniques were applied over it to retrieve the information. But, OCR deteriorates severely when the images were of poor quality or complicated layout. Motivated by this observation, some retrieval methods with the ability to tolerate the recognition errors of OCR have been researched later (Ishitani 2001, Ohtam et al 1997). Additionally, some methods were reported to improve retrieval performance by using OCR candidates (Kameshiro et al 1999, Katshuyama, 2002). Even though, many methods were reported in OCR for candidate correction, keyword spotting could bypass OCR in information retrieval because it is very particular in answering whether the document image contains the words which are of interest to the user. Moreover, keyword spotting could achieve a higher precision, recall and speed than OCR (Lu and Tan 2004a).

Therefore, researchers slowly moved to keyword spotting technique, which retrieves information from document images without converting the entire document image into text (ASCII code) representation and the text is identified at word level using the properties of image (Doermann 1998).

Keyword spotting approaches are broadly classified as Character based keyword spotting and Word based spotting and sometimes there is an integration of the two. In character based spotting, features are represented at character level. Each character is encoded with a character shape code and they are joined together to represent a word. In Word based spotting, features are extracted at the word level directly instead of character level which is insensitive to character segmentation error. From this viewpoint, character based word spotting methods can be classified as Character Shape Analysis
methods, N-gram based character object methods and pixel based character matching methods. Word based spotting can be classified into Statistical methods, Hausdorff methods, Coarse feature based methods, Primitive String (LPRS) methods and Geometric Feature Graph (GFG) methods as in Figure 2.3. In this thesis, an integrated scheme of character and word based feature extraction has been attempted to spot the keywords in documents.

![Figure 2.3 Classification of Keyword Spotting methods](image)

2.4.1 Character Level Feature Representation

Character level feature representation has been done under feature level (language dependent) and pixel level (language independent) with the three techniques, Character Shape Analysis methods, N-gram based character object methods and pixel based character matching methods as represented in Figure 2.3.
2.4.1.1 Character Shape Analysis

A single character, which has been encoded in a character shape code alphabet, has been referred as character shape codes (Reynar et al 1995). A word, which has been encoded with character shape codes, is called as word shape token.

In character shape analysis, on each text line, four horizontal lines define three significant zones (Reynar et al 1995). Area between the base line and the top of characters was called as x-zone. Area above the x-zone was called as ascender zone and the area below the x-zone was called as Descender zone. The text line has been further divided into character cells by vertical boundaries. Each cell contains the connected components of a single character image.

Initially, character shape code was developed as two-bit code (Tanaka and Torii 1988) in which the high order bit encodes x-height (Ascenders) and non x-height characters and the low order bits indicates whether the character form has exactly one or more than one crossing of horizontal line midway in the elevation between the baseline and x-line. First bit divides the character set into two groups (Ascenders and Non-Ascenders) whereas the second bit was font dependent.

Later, three character shape codes representing three groups of characters were introduced to provide a better discrimination among characters (Schurmann et al 1992, Sinha 1990). Three shape codes represent the characters, which lie in Ascender zone, x- zone and Descender zones. This method has been further explored into five shape codes named as $V_0(A,g,x,j,i)$ to distinguish the character j from i (Spitz 1993). Still this technique suffers due to the character confusions that arise within the same group.
The most frequent character confusions can be seen due to the confusions between e and (s, r, a, o, n) in words, and partition of the set of x-height characters into two sets, one of which contains e and the other of which contains (s, r, a, o, n) reduces the confusion further. Later partitioning of x-height characters into two sets was achieved by the criteria (Spitz 1995), which identifies the presence of a deep east concavity within the middle third of the character. This partition result in the new mapping scheme of six character shape codes named as $V_1$, by separating ‘c’$’s and ‘e’$’s from the rest of the x-height characters and maps to the character shape code ‘e’. Subsequently, a further shape code ‘n’ was added to distinguish the letter n from all the x-height characters based on the presence of a single southward concavity. This seven character shape coding (Reynar et al 1995) named as $V_2$, reduces the ambiguity associated with word shape token by increasing the character shape code alphabet.

Word spotting and searching initiated with the character shape code $V_1$, produced an average rate of 82% of Precision and 72% of Recall (Spitz 1995). The false positives arose from the mapping of multiple words in the English language to a particular tokenized representation and negatives were largely due to the failure of the character splitting algorithm to effectively split the touching characters in degraded document images. Formation of Word shape tokens using the shape codes $V_0$ and $V_2$ has been attempted for indexing and searching in document image sets by Spitz later (Smeaton and Spitz 1997).

Though several character shape coding schemes have been reported, few of them can deal with camera images of documents that were degraded by the perspective distortion. Therefore a novel keyword spotting scheme was attempted to locate keywords within document images captured by a digital camera (Lu and Tan 2007). Vertical and Horizontal text directions were first identified in a camera image of document and the shape of word
images were captured by using three perspective invariants such as character holes, Water reservoirs and character Ascenders, Descenders. These three invariant features transliterate each character into a digit sequence of six dimensions and convert each word image into a word shape code. Since this technique was character based, which converts each character into a digit sequence of dimension six, the coding would be wrong if one or more spelled characters are touching or broken.

Since character shape coding schemes assign a character shape code to each character, better discrimination among the characters would be possible only if the number of shape codes gets increased. Reduced number of shape codes has lower computational complexity but results in ambiguity among the characters in the same class. Moreover, the shape codes are language dependent and cannot be performed over other languages. In order to extract features from the character image objects, irrespective of their language, pixel based feature extraction from character image objects and similarity measurement using N-gram method has been proposed.

2.4.1.2 N-gram Methods

This is a pixel based feature extraction technique, which extracts the pixel properties of character image objects, and constructs document vectors. Later vectors are measured for their similarity using N-Grams. This technique has been proposed to retrieve information from multilingual document images.

Tan proposed a new method of text retrieval from document images using a similarity measure based on an N-gram algorithm (Tan et al 2000, 2002). Here character image objects were extracted from document images based on connected component analysis. Each character object was represented by two object features, namely, Horizontal Traversal Density
(HTD) and Vertical Traversal Density (VTD). HTD consists of a vector whose elements denote the numbers of line segments when the character was scanned horizontally line by line from top to bottom. Similarly, VTD denotes another vector obtained from vertical scanning from left to right. An n-gram vector was constructed for each document based on character features. Based on n-gram vectors, classifier measures the similarity for document retrieval. The n-gram vector then serves as a text similarity measure for document retrieval.

The above method was improved further towards different fonts and noisy documents by using word shape descriptors such as local extrema points, to obtain the similarity between document images (Tan et al 2003). Here document images were segmented into word units and vertical bar patterns were extracted from these word units. Here, word has been represented by a list of vertical bars contributed by its characters. All vertical bar patterns were used to build document vectors. Finally, feature vectors have been used to calculate the similarity of document images by calculating their scalar products. Due to its faster processing, this method offers a speedier way of retrieving scanned images based on content similarity.

This method rise a heavy ambiguity between words and therefore it was suitable for imaged document similarity estimation, but not able to handle spotting with a few keywords as queries. Therefore, pixel based feature extraction of character image objects has been explored further on Chinese document images using Hausdorff distance, which measures the similarity between two character image objects for query processing.

2.4.1.3 Character based Hausdorff Methods

Many approaches have been attempted to spot word images in English but relatively very few for Chinese document images. But, character
based word spotting scheme was attempted in Chinese document images to search user specified words or phrases by Lu and Tan (2002b, 2004b). Here, connected components were detected and merged to bound characters or elements in the word images. For query processing, a template image was generated from the selected initial character, and then matched with each bounded character image in the document based on the character matching method. Performance Evaluation yielded a precision rate of 84%.

Therefore, in all the above methods, pixel based or feature based, character segmentation is a necessary step before downstream processes. In many cases, especially with document images of poor quality, it is not easy to separate connected characters and they are sensitive to character segmentation error. Unfortunately, all the above methods could not deal with touching characters. Therefore, word level feature representation has been proposed to handle this issue.

2.4.2 Word Level Feature Representation

To avoid the difficulties of separating, touching characters (adjacent characters touch with one another) in a word image, segmentation-free methods have been researched in some document image retrieval systems, in which image matching at the word level has been utilized (Lu and Tan 2004a). A segmentation-free word image matching approach treats each word object as a single, indivisible entity, and attempts to recognize it using features of the word as a whole.

Word level feature representation and matching could be performed either at the pixel level or feature level. Pixel level processing is language independent whereas feature level processing is language dependent. Pixel level representation and matching has been performed by Statistical methods, Hausdorff methods and Coarse feature based methods whereas the feature
level representation has been addressed by Primitive String technique and Geometric Feature Graph (GFG) Extraction technique.

### 2.4.2.1 Statistical Methods

Statistical methods identify the keywords with a priori model trained for each character. Initially, a segmentation and recognition free approach for information retrieval has been developed by Chen et al (1993). He first identified candidate lines of text using morphology and extracted shape information such as upper and lower contours of each word, from the normalized lines of text. A keyword Hidden Markov Model (HMM) has been created from a series of appropriate character HMMs and a non-keyword HMM was based on context-dependent sub-character models. The character HMMs were trained using data from different fonts.

A subsequent paper (Chen et al 1995) described a simplification of this process, which used vertical character alignment information and a single model for each character. Many systems addressed Hidden Markov models to identify the keywords in document images. But, this model could work only for the specific cases or for the predominant fonts in which the character HMM has been created.

In order to overcome the above shortcomings, Chen and Bloomberg (1996) used word shape information and a voting technique to perform matching of keywords, without segmentation. The approach was based on features including blanks, horizontal strokes, vertical strokes, ovals and bowls extracted from a contiguous line of text. Later, a priori model has been constructed that specifies which words should contain which patterns. At runtime, the same features were extracted from a line of text and a voting scheme provides keyword hypotheses.
Statistical methods could process only predefined keywords and a pretraining procedure was mandatory. A lexicon must be an available tool for resolving the ambiguity. Since this method was lexicon dependent, Hausdorff similarity measure based word level matching has been proposed.

2.4.2.2 Word based Hausdorff Methods

As a low-level matching, Hausdorff distance has been used for word image matching, which seems to be very simple and insensitive to changes of image characteristics. Hausdorff distance tells the similarity between pixels by measuring how two pixel subsets are far from each other.

As a pixel level processing, Yue Lu attempted a word image matching procedure (Lu et al. 2001a) which matches the user specified word and the word images extracted from document images using Weighted Hausdorff distance measure. Here word images were normalized into template size image and divided into different parts, namely the ascender, descender and middle zone with different weights assigned to them. Later, a weighted Hausdorff distance was used to measure the distance between the template image and document image and documents were retrieved when the distance was less. Later, Yue Lu extended the above word image matching procedure (Lu et al. 2001b) to CCITT Group 4 compressed document images.

In addition, Yue Lu utilized the same technique to compare the similarity between CCITT Group 4 compressed document images (Lu and Tan 2003). Although the system can cope with any word the user specifies, this could not deal with the problem of partial matching. Therefore, coarse features based methods for word level matching arises to handle partial matching even in pixel level processing.
2.4.2.3 Coarse Features based Methods

Initially, pixel based coarse features was introduced in Telugu document images (Jawahar et al 2004a) for information retrieval. Three coarse features such as word profiles (upper profile, lower profile, projection profile, ink transition), structural features and transform domain representations were addressed to indicate the features of a word image. Structural features involve normalized moments, first order moments and statistical moments such as the mean, standard deviation and skew of the word images. Transform domain representations include Fourier coefficients. Feature values were normalized such that the word representations become insensitive to variations in size, fonts and degradations. Similarity between the words has been identified using structural similarity of word images obtained by comparing the shapes and a sequence alignment score has been computed by Dynamic Time Warping procedure (Rath and Manmatha 2003). When a textual query has been given to the system, it gets converted into and image by rendering and features are extracted from the word image and measured using distance measures. This technique has also been applied to promote keyword search (Balasubramanian and Jawahar 2006) within the graphics stream of PDF files.

Later, this coarse features extraction system was extended to indexing of word images in documents (Jawahar et al 2004b), which also addresses the issue of variations at the end of the word forms. A Word form variation occurs at the end of the word such as direct-directed. Subsequently, search was extended (Balasubramanian et al 2006) to allow the user to retrieve all the documents related to their queries in any of the Indian languages. Cross lingual search (English and Hindi) was implemented by the transliteration among Indian languages and a table-lookup translation for other languages. Here search of a query word fetches relevant pages from
both the scripts. Coarse feature extraction system was extended further (Pramod Shankar and Jawahar 2006) to annotate the document images with the textual keyword. This was particularly suitable for the document images, where the knowledge of the vocabulary provides with the possible annotations.

Since coarse features could work well across the scripts, it was adopted for three languages (Hindi, Telugu and Bengali) by Harit et al (2005b) to index the documents. Here envelopes (Word Profiles) are used to represent the global and local shape properties of the word. The envelope curve is partitioned by the principle axis of the word image into two halves, upper envelope (upper profile) and lower envelope (lower profile). Local information on the envelope curves (Peaks and valleys) were identified by placing a 7*7 grid over the envelope and assigning feature values for every cell in the grid. Therefore, the word image has been represented with two feature strings, one for the upper and the other for the lower envelope curve, each of which comprises of 49 characters. Edit distance was used to compare the corresponding envelopes of two words. Even though this method can address the variations in word forms, this could address only the variations at the end of the word. Therefore, a complete partial matching cannot be achieved here. This issue could be solved only by the feature level processing.

All the above pixel level methods addressed for word level feature representation are language independent techniques and could work across different languages. But promising precision and recall cannot be achieved over a large corpus, since they require appropriate training sets. Since, no language specific features have been involved, pixel level feature vectors are very difficult to get indexed. On the other hand, feature based methods are advantageous, since they are easy to form query, easy to index and training free.
2.4.2.4 Primitive String Methods

Primitive String generation for word images is a variation of character shape analysis. Character codes are not assigned to the characters individually. Instead, features are represented at the word level for image objects. This technique mainly deals with the character segmentation problem especially connected characters. Procedure of computing the word image code is complicated when compared to character shape analysis methods but word shape coding eliminates the ambiguity among words easily.

In primitive string generation, an initial attempt was made by Yue Lu and Tan to represent the word images by a feature string as a whole, (Lu and Tan 2002a) instead of representing in character codes. Here primitive string was generated by scanning the word image column by column and giving a feature code for each column. Vertical Strokes, Long and short vertical strokes, Upper long vertical strokes and lower long vertical strokes have been used as features to represent feature codes. Search process was initiated over it by synthesizing the feature string for the user specified query word according to the character sequence and matching the relevant documents. Subsequently, this technique (Zhang et al 2004) was applied for a web interface system, which retrieves documents from digital libraries.

This technique was able to deal with most commonly used fonts, but it cannot handle italic fonts and uncommon fonts. This occurs due to the shortcoming of features used to represent the word image. Since features were obtained by vertically scanning the image column by column; this is unable to accommodate italic fonts. In addition, due to the exploration of limited features, more fonts were not accommodated by this technique.

In order to overcome the above shortcomings, feature code for word image objects (Lu et al 2004a) has been represented as a Left to Right
Primitive String (LPRS) by Yue Lu. A word was explicitly segmented, from the leftmost to the rightmost, into discrete entities. Each entity has been represented using two tuple attributes Line or Traversal attribute and Ascender-and-Descender attribute. Line or Traversal attribute deals with the presence of vertical lines and black pixel transition in every column whereas Ascender-and-Descender attribute deals with the presence of ascenders and descendents. In retrieval process, user’s query would be converted into the same feature encoding process. This technique could work across the word image objects of various sizes, fonts, spacing and partial match. LPRS technique has also been extended by Lu et al (2004b) to search words in PDF files of imaged documents, using Acrobat SDK.

Later, LPRS technique was applied by Lu and Tan (2004a) to address two issues in document image retrieval: matching partial word images and similarity measurement between documents. Here, each word image has been represented by a primitive string (LPRS string), which could tolerate font differences, styles and touching characters. Then, an inexact string matching technique was utilized to measure the similarity between the two primitive strings generated from two word images. Even though this technique tolerated variations in styles and typefaces, typesetting in matching score but depend heavily on threshold setting for similarity measurement.

To simplify primitive string generation, a new word shape coding scheme has been proposed (Lu et al 2008) to capture the document content through annotating each word image by a word shape code. In particular, word images were annotated by using a set of topological shape features including character ascenders/descenders, character holes, and character water reservoirs. With the annotated word shape codes, document images could be retrieved by either query keywords or a query document image.
This Primitive String generation technique is insufficient to represent Devnagari script since it mainly concentrates on the characteristics of Roman script. This technique fails to exploit the structural characteristics of Devnagari script. Therefore Geometric Feature GFG technique has been developed to represent the features at word level in Hindi images.

2.4.2.5 Geometric Feature Graph Method

Geometric Feature Graph (GFG) representation for word images was introduced by Harit et al (2001) by exploiting the typical structural characteristics of Devnagari based scripts. GFG, a graph based representation scheme for encoding structural relationship (Chaudhury et al 2003) between the shape primitives was used by Harit to characterize the word images. Here the nodes in the GFG were end points or junction points in the word image and the branch between two nodes represent the generic shapes (horizontal line, vertical line, semicircle, ellipse etc.) connecting two node points in the word image. Here word images undergo a thinning algorithm to obtain its skeletal representation and the skeleton tracing algorithms traces both the connected paths (branches) in the word images and terminal points (nodes) and label them to produce the symbolic descriptor for the word image. Here positioning and labeling of nodes was independent of the traversal path. Query word image also undergo the same process defined above to match the documents for retrieval.

Subsequently, GFG technique was adopted in the development of a system named Heritage + (Harit et al 2004) which deals with document images as a distinct media type and implements tools and techniques for browsing and querying document images along with other media elements like audio, video sequences and images. Later GFG technique was improved (Harit et al 2005a) to achieve compressed strings instead of a long symbolic descriptor for every word image with two shape primitives line segment and
curve segment. Subsequently, computational speed up to the above GFG extraction algorithm was achieved (Harit et al 2007) through a modification of the primitive identification technique. Extraction of the length and orientation attribute of Line and Curve primitives has been computed at each pixel using local tangent information.

However, feature extraction techniques discussed above are specific and language dependent. Consequently, for Tamil document images, need for information retrieval arises in the context of digitizing Tamil documents from ancient and old era to the latest. No specific word spotting technique reported above could be applied to the Tamil language directly since the features of Tamil language (i.e.) shape of the Tamil characters, vary from other languages.

2.4.3 Previous Works in Text Image Understanding

Tamil text recognition systems developed so far have their own constraints over the font faces or sizes of letters. Either it is restricted to a particular size or a range of font sizes. In addition, these text recognition systems (Subramanian and Kuberan 2000, Krishnamoorthy 2002) suffer weaknesses to discriminate a group of closely resembling characters in the character set (Seethalakshmi et al 2005). This also necessitates post processing or spell check to correct the errors occurred in recognition.

These outstanding problems in Tamil text recognition systems motivated us to the idea of developing LR-TB-FS technique, to retrieve information from Tamil document images. The idea is based on the assumption that the technique devised would extract word images across various font faces and font sizes instead of training the shapes of the characters. Here the basic features are extracted by traversing through the vertical centroid area and horizontal zones of characters to record their black and white disposition rates and this could work well across fonts.