Devising Interactive Access Techniques for Indian Language Document Images

Santanu Chaudhury* santanuc@ee.iitd.ernet.in
Geetika Sethi geetu_sethi@yahoo.com
Anand Vyas anand_ju@rediffmail.com
Gaurav Harit gharit@ee.iitd.ernet.in

Department of Electrical Engineering
Indian Institute of Technology, Delhi
Hauz Khas, New Delhi 110016, INDIA

Abstract

A large volume of legacy documents in Indian languages exist only in paper form. Web based interactive access techniques for images of these documents can ensure wider dissemination and easy availability. In this paper, we have proposed an access mechanism based on word based indexing and personalized annotation. The word based indexing scheme exploits typical structural characteristics of Indian scripts. We have combined this word indexing technique with personalized annotation based hyperlinking and query scheme for providing an interactive access interface to a collection of Indian language documents.

1. Introduction

A large volume of legacy documents in Indian languages exist only in paper form. Web based interactive access techniques for images of these documents can ensure wider dissemination and easy availability. In this paper, we have presented a new access scheme based on word image based indexing and personalized annotation. Our scheme has been devised with an automatically derived XML based representation of document images [9]. This approach, however, does not involve use of OCR because reliable OCR's for Indian languages are not yet easily available.

The problem of word based document image indexing has received attention in the past [8]. DeSilva and Hull [6] have addressed the problem of detecting proper nouns in document images. Some interesting work has been done on the problem of searching for keywords in document images using only image properties. The approach of Chen et al. [2] is segmentation- and recognition- free. Chen et al. first identify candidate lines of text using morphology and extract shape information from normalized lines of text. The upper and lower contours of each word are identified and used, along with the auto-correlation between columns of pixels, as input to a Hidden Markov Model (HMM) for key word spotting. Related components of the system are described in more detail in [3], [4] ,[5] and [10]. DeCurtins and Chen [7] use word shape information and a voting technique to perform matching of keywords, also without segmentation. The approach is based on features including blanks, horizontal strokes, vertical strokes, ovals and bowls extracted from a contiguous line of text. These techniques do not derive a symbolic representation of the keywords so that standard IR (information retrieval) techniques can be used with document images. Further, these techniques do not always exploit script specific characteristics for developing the indexing scheme. Exploiting language specific characteristics, a key word recognition scheme for oriental languages was proposed in [14].

In this paper, we have suggested a new word based indexing scheme which exploits typical structural characteristics of Devnagari based scripts. We have proposed a novel symbolic representation scheme for encoding structural relationship between shape primitives characterizing word images. The string based symbolic representation can be used for implementing useful IR techniques. We have combined this word indexing technique with personalized annotation based workspace construction scheme for providing an interactive access interface to a collection of Indian language documents. The suggested individualized organization of document images for query processing and hyperlinking is also a novel contribution of this work.

2 Word Based Indexing

We have developed an indexing scheme which uses a symbolic representation derived from word images. Indexing words are identified in the document collection through an interactive interface. Query words are also indicated through a similar interface. We shall describe the interface

*Author for correspondence

Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)
0-7695-1960-1/03 $17.00 © 2003 IEEE

in the next section.

2.1. Geometric Feature Graph Extraction

For the purpose of indexing, word images are represented in the form of geometric feature graphs (GFG). Geometric Feature Graph (GFG) is a graph based representation of the features extracted from the image of the word. The GFG used for the present work has following specifications:

- The nodes in the GFG can be either the end points or junction points.
- The branch between any two node represents the type of generic shape connecting the two node points in the character image. The type of connection can be one of those shown in fig 1.

The GFG's of word images indicate basic connectivity relation between the shape primitives which form the word. The set of shape primitives chosen can adequately represent shapes of characters of Devnagari and other Devnagari based scripts. Classification of the connecting branches into one of the predefined types makes the representation scheme invariant to certain classes of geometrical transformations. For example, in a given GFG, representation of the features of the component characters is invariant to the size of the font. The symbolic classification of the type of the connector also makes the representation scheme invariant to limited deformations due to change in the type of fonts and non-linearities due to errors in the imaging process. Experiments also corroborate the above expectation. Features used for the graph representation have been obtained after thinning the word image.

The GFG formation involves the following steps:

Graph Formation In the first phase extra pixels are removed and breaks are joined in the thinned image using error-correction heuristics for constructing a tentative GFG model.

Graph Refinement The second pass is for labeling nodes and branches of the GFG.

A skeletal representation of the segmented word image is obtained by a thinning algorithm. The *'skeleton obtained'* may have extra pixels and breaks.

In the first phase GFG is constructed by following the skeleton of the thinned image of the character or word. The skeleton tracing algorithm traces the connected paths in the words and identifies terminal points and multiple connected pixels as potential nodes of the GFG. This algorithm is a modification of the well known stack based contour tracing algorithm. The direction of tracing is very important as the indexing scheme requires an invariant ordering of the nodes of the GFG. So while checking for the neighboring unvisited black-pixels the order in which the pixels are considered are P1, P7, P3, P5. In fact pixels are pushed into the stack in that order. The pixel P0 will be considered only if P1 and P7 are not black. Similarly, P2, if P1 and P3 are not black; P4, if P3 and P5 are not black; P6, if P7 and P5 are not black. Since the last pushed element is popped and traced, this makes sure that the direction it will follow will be first down , then left, then right and finally top at the end.

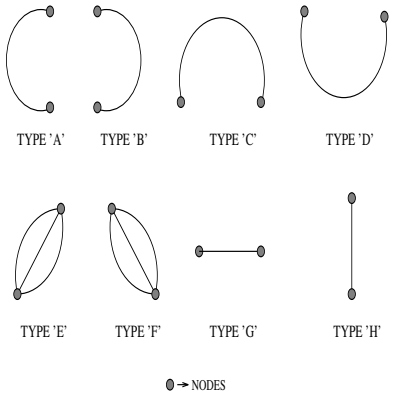


Figure 1. Different types of Connecting Elements between Nodes

P2	P1	P0
P3	P	P7
P4	P5	P6

In the case of a Devnagari word the starting point of the algorithm will be the left edge of the top horizontal bar (assuming portrait orientation). The algorithm will trace all the segments of the first character and then only it will go to the next character. The top ligatures will be traced only at the end.

In the second pass, branches of the GFG are labeled and an unambiguous ordering of the nodes is achieved. Branch labeling is done using the terminal points xt_1, yt_1, xt_2, yt_2 of the connected nodes and the $x_{min}, y_{min}, x_{max}, y_{max}$ from the path traversed from the one node to other.

Let xt_1, yt_1, xt_2, yt_2 are the coordinates of the two connected nodes. Let x_{min}, y_{min}, x_{max}, y_{max} be the minimum and maximum values of the coordinates obtained while traveling from node1 to node2. Now,

$$x_{min-diff} = min(\mid xt_1 - x_{min} \mid, \mid xt_2 - x_{min} \mid)$$

$$x_{max-diff} = min(\mid xt_1 - x_{max} \mid, \mid xt_2 - x_{max} \mid)$$

$$y_{min-diff} = min(\mid yt_1 - y_{min} \mid, \mid yt_2 - y_{min} \mid)$$

$$y_{max-diff} = min(\mid yt_1 - y_{max} \mid, \mid yt_2 - y_{max} \mid)$$

A set of rules based on these parameters is used to identify generic nature of the connectors which does not change because of the similarity or affine transformations [11]. Small deformations due to change in font type also does not alter these generic properties.

As mentioned earlier, nodes are either end points or junction points of branches. The traversal algorithm traverses the skeletonized image in a specific order of direction. The path of traversal may be altered at junction points depending upon the number and position of neighbors at the junction. This results in change in relative positioning of nodes. So after allocating nodes, they are ordered according to their x-y coordinates, making node positioning independent of the traversal path. Fig 2 shows the GFG of a word image obtained after the graph extraction.

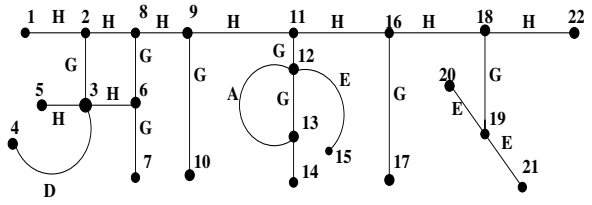


Figure 2. GFG of a Hindi word - Aakar

2.2 Indexing

The indexing string in the form of symbolic descriptors is obtained by traversing the word GFG in depth first fashion. For example the symbolic descriptor string for word image shown in Fig. 2 will be **HGDHHGHGHGH-GAGGEHGHGEEH**. After the extraction of the symbolic string for the word images the problem basically reduces to that of string based indexing. To accommodate the possibility of matching to root words and image processing errors we need an indexing scheme with the facility for approximate match. The indexing problem is formulated as: given a *text* of length *n*, and pattern of length *m*, retrieve all the text segments(or "occurrences") whose *edit distance* to the pattern is at most *k*.The *edit distance* between two strings is defined as the minimum number of character insertions, deletions and replacements needed to make them equal. We define the "error level" as $\alpha = k/m$. Here text will be the symbolic descriptors for all the word images in the database and pattern will be the descriptor for the query word image.

We have used suffix tree for indexing. In this approach, the pattern is partitioned into sub-patterns which are searched for. All the occurrences of the sub patterns are latter verified for a complete match. The goal is to balance between the cost to search in the suffix tree(which grows with the size of the sub patterns) and the cost to verify the potential occurrences (which grows when shorter patterns are searched). We used Ukkonen's linear-time algorithm [12] for construction of the tree. When the query image comes in,it is processed as above to get the symbolic descriptor string corresponding to the image. Now, for searching the pattern in the suffix tree, we sub divide the pattern. The size of the sub-patterns considered is a function of length of the pattern ($> 0.4*$ length of query pattern). All the strings having the sub patterns are selected for verification. Verification is done by calculating the edit distance between the pattern and each of the string selected for verification. All the strings having edit distance within a given threshold are considered for indexing on to documents. The threshold is also taken as function of the length of the query pattern. Use of suffix tree based approximate string matching based indexing for document images is a novel contribution of this work.

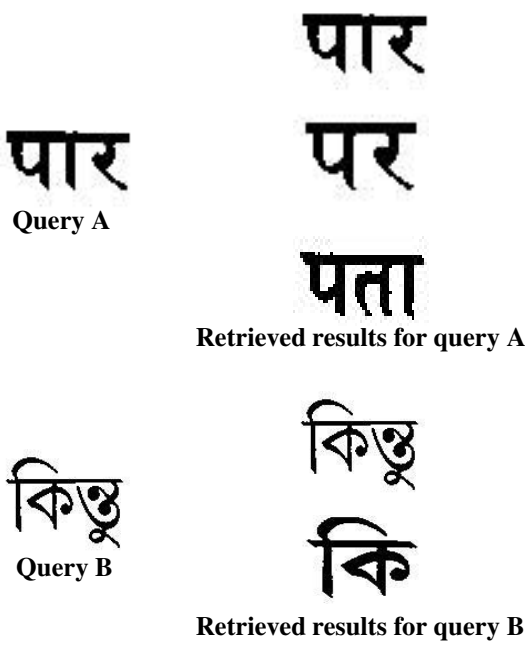


Figure 3. Examples of Images retrieved using Symbolic Descriptors Matching

We have done experiments with Hindi and Bengali words. The database contained 10,000 words. Fig. 3 shows two examples of retrieved words. Fig. 4 shows results for precision. The results are the average of a total of 100 queries, with λ (edit distance allowed) as the evaluating parameter. The database for different scripts have been maintained separately. We have used the script recognizer[1] for organizing document collections according to the script. Experimental results show that this technique combined

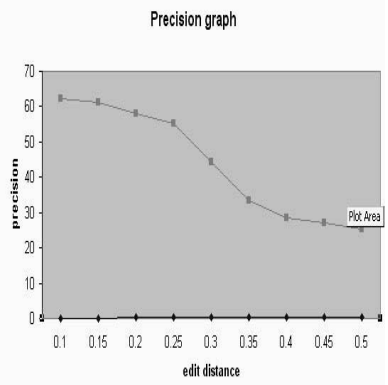


Figure 4. Average Precision values for 100 queries

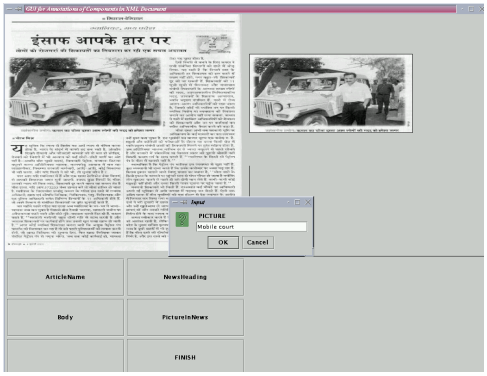


Figure 5. A snap shot of annotation GUI

with other semantic information can be used effectively for accessing document images.

3. Document Image Access in Personalized Workspace

The original document image is converted to a XML representation using model guided segmentation [9] which results in extraction of component blocks of semantic significance. In XML, structured annotations can be provided by means of tags, or Elements, representing the information by virtue of their names and the attribute/value pairs associated with the element. These XML representations and actual image components are maintained separately. Consequently, an image or a collection of images can be associated with individualized descriptors or annotations generating virtual workspace for individual users by reusing the same image components. An user is provided with a variety of tools like annotation based indexing or hyperlinking, image similarity based hyperlinking or querying and word image based indexing of components (as described in the previous section), for designing access schemes for a set of documents. These access modalities can exploited by him/her or a group of users selected by him/her for referring these documents. In other words, using these tools an user can assume the role of a reference librarian for a document image collection. Also, there can be more than one such librarian organizing differently the same set of documents according to his/her own perspective.

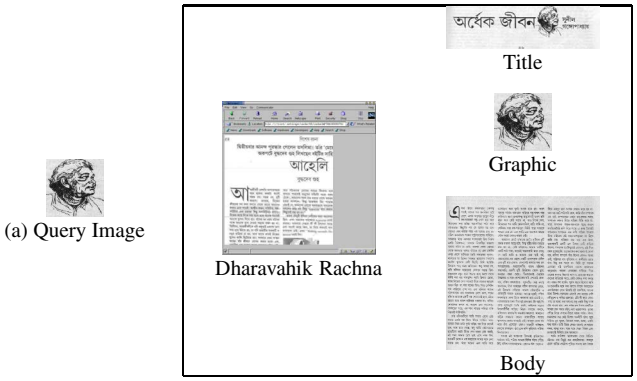
Annotations are textual descriptions associated with logical components of a document. A GUI, using JAVA Swings, was developed to make the annotation task easy and effective. Similar GUI is used for indicating indexing words in the document components for constructing word image based indices (see Fig. 5).

3.1 Hyperlinking

We may use links to define relationship among similar documents or to define a sequence in which documents can be navigated. The specification for XML linking is known as XLink. HTML links are fairly inflexible compared with XLink due to the drawbacks that they are embedded in the source document, allow navigation in only one direction and can connect only two resources. In the present scheme out-of-line Extended links (that is, links specified in a document other than the source document) were used which overcome the above mentioned drawbacks. A special type of <xlink:extended> element is used to indicate that out-of-line links exist for a particular document. Thus linking features of XML technology provide a mechanism to present a generalized hyper-linking scheme among various XML documents, and in turn among document images, based on attributes in individualized workspace. For example, certain articles in the magazines are published as a series of stories. Various episodes of such serials can be hyper linked using XLink feature and made available as a result of user query. An example of this kind of a Generalized Hyperlinking scheme has been shown in Figure 6. Such hyperlinks are constructed automatically using annotations and image based similarity using features of image components (e.g. Color Histogram, Gabor or Wavelet filter based feature). For the given example, different instances of the serial in the collection were linked by detecting occurrence of the same image in the title block.

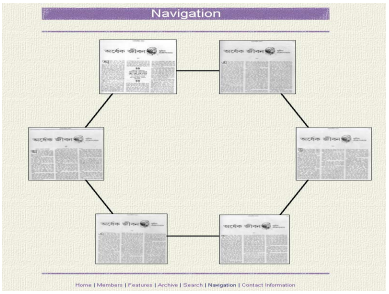
3.2. Query Processing

Apart from supporting queries based on text, word images, the present system supports query by image example. For this purpose wavelet based features and similarity measures as discussed in [13] are used. A result of image-similarity based query are shown in Fig. 7.



(a) Query Image

(b)



(c)

Figure 6. This figure illustrates the automatic generation of generalized hyperlinks. Part (a) shows a query image. This image is matched with all the image components of all the documents in the database. The documents which have images similar to the query image (see part (b)) are retrieved. All such documents are hyperlinked. Part (c) shows navigation of hyperlinked episodes of Bengali story serial. The user can click on any icon and see that particular page.

Query Image A Match 1 Match 2

Figure 7. Results: Retrieval of Images similar to a Query Image

4. Conclusions

In this paper we have proposed a new scheme for word image based indexing of document images. We have also suggested a novel scheme for hyperlinking and annotating document images for personalized access. The indexing scheme and personalized access mechanisms are novel contributions of this work.

References

[1] S. Chaudhury and R. Sheth. Trainable script identification strategies for indian languages. *Proceedings of the International Conference on Document Analysis and Recognition*, pages 657–660, 1999.

[2] F. Chen, L. Wilcox, and D. Bloomberg. Detecting and locating partially specified keywords in scanned images using hidden markov models. *Proceedings of the International Conference on Document Analysis and Recognition*, pages 133–138, 1993.

[3] F. Chen, L. Wilcox, and D. Bloomberg. Word spotting in scanned images using hidden markov models. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1–4, 1993.

[4] F. Chen, L. Wilcox, and D. Bloomberg. A comparison of discrete and continuous hidden markov models for phrase spotting in text images. *Proceedings of the International Conference on Document Analysis and Recognition*, pages 398–402, 1995.

[5] F. Chen and M. Withgott. The use of emphasis to automatically summarize a spoken discourse. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 229–232, 1992.

[6] G. L. De Silva. Proper noun detection in document images. *Pattern Recognition*, pages 311–320, 1994.

[7] J. DeCurtins and E. Chen. Keyword spotting via word shape recognition. *Proceedings of the SPIE Document Recognition II*, pages 270–277, 1995.

[8] D. Doermann. Indexing and retrieval of document images: A survey. *Computer Vision and Image Understanding*, pages 287–298, 1998.

[9] G. Harit, S. Chaudhury, P. Gupta, N. Vohra, and S. Joshi. A model guided document image analysis scheme. *Proceedings of the International Conference on Document Analysis and Recognition*, 2001.

[10] J. Kupiec, J. Pedersen, and F. Chen. A trainable document summarizer. *Proceedings of the 18th Annual ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 68–73, 1995.

[11] G. Shethi. Providing interactive access to legacy documents. *M.Tech Thesis, I.I.T,Delhi*, 2002.

[12] E. Ukkonen. Finding approximate patterns in string. *Journal of Algorithms*, pages 132–137, 1985.

[13] J. Wang, G. Wiederhold, O. Firschein, and X. S. Wei. Content based image indexing and searching using daubechies wavelets. *Digital Library*, pages 311–328, 1997.

[14] J. Zhu, T. Hong, and J. J. Hull. Image-based keyword recognition in oriental language document images. *Pattern Recognition*, pages 1293–1300, 1997.