Retrieval Of Information In Document Image Databases Using Partial Word Image Matching Technique

Seema Yadav , Dr. Sudhir Sawarkar

Abstract-With the popularity and importance of document images as an information source, information retrieval in document image databases has become a challenge. In this paper, an approach with the capability of matching partial word images to address two issues in document image retrieval: word spotting and similarity measurement between documents has been proposed. Initially, each word image is represented by a primitive string. Then, an inexact string matching technique is utilized to measure the similarity between the string generated of the query word with the word string generated from the document. Based on the similarity, we can find out how a word image is relevant to the other and, can be decided whether one is a portion of the other. In order to deal with various character fonts, a primitive string which is tolerant to serif and font differences to represent a word image has been used. Using this technique of inexact string matching, our method is able to successfully handle the problem of heavily touching characters. From the experimental results on a variety of document image databases it is confirmed that the proposed approach is feasible, valid, and efficient in document image retrieval.

Index Terms—Document image retrieval, partial word image matching, primitive string, word searching,.

1.**Introduction**

MODERN technology has made it possible to produce, process, store, and transmit document images efficiently. In an attempt to move toward the paperless office, large quantities of printed documents are digitized and stored as images in databases. If we look through the documents stored in digital libraries for eg., digital library of our university ,online storage of books, students theses etc. are simply scanned and archived in image form that cannot employ the current powerful search engines over text. The popularity and importance of document images as an information source are evident. As a matter of fact, many organizations currently use and are dependent on document image databases. However, such databases are often not equipped with adequate index information. it difficult to retrieve user-relevant information from image data than from text data. Thus, the study of information retrieval in document image databases is an important subject in knowledge and data engineering. Millions of digital documents are constantly transmitted from one point to another over the Internet. The most common format of these digital documents is text, in which characters are represented by machine codes. .To make billions of traditional and legacy documents available and accessible on the Internet, they are scanned and converted to digital images using digitization equipment. Although the technology of Document Image Processing (DIP) may be utilized to automatically convert the digital images of these documents to the machine-readable text format using Optical Character Recognition (OCR) technology, it is typically not a cost effective and practical way to process a huge number of paper documents. One reason is that the technique of layout analysis is still immature in handling documents with complicated layouts. Another reason is that OCR technology still suffers inherent weaknesses in its recognition ability, especially with document images of poor quality.

In recent years, there has been much interest in the research area of Document Image Retrieval (DIR) [1], [2].DIR is relevant to document image processing (DIP), but there are some essential differences between them A DIP system needs to analyze different text areas in a page document, understand the relationship among these text areas, and then convert them to a machine-readable version using OCR, in which each character object is assigned to a certain class. DIR system provides an answer of "yes" or "no" with respect to the user's query which is of interest, .Moreover, words, rather than characters, are the basic units of meaning in information retrieval. Therefore, directly matching word images in a document image is an alternative way to retrieve information from the document. In short, DIR and DIP address different needs and have different merits of their own. The former, which is tailored for directly retrieving information from document images, could achieve a relatively higher performance in terms of recall, precision and processing speed.

1.1 Related Work

A method described by Chen and Bloomberg automatically select sentences and key phrases to create a summary from an imaged document without any need for recognition of the characters in each word. Liu and Jain proposed a method of similarity measure for forms that is insensitive to translation, scaling, moderate skew, and image quality fluctuations. Niyogi and Srihari [3] described an approach to retrieve information from document images stored in a digital library by means of knowledge-based layout analysis and logical structure derivation techniques, in which significant sections

Seema Yadav,*MGM college of Engg. Kalamboli,*,
Email :contactseema2000@yahoo.com
Dr. Sudhir Sawarkar,*Head of computer department*
Datta Meghe COE, Airoli
Email:sudhir_sawarkar@yahoo.com

of documents, such as the title, are utilized. Tang et al. [4] proposed methods for automatic knowledge acquisition in document images by analyzing the geometric structure and logical structure of the images. He et al. [5] proposed an index and retrieval method based on the stroke density of Chinese characters. derivation techniques, in which significant sections of documents, such as the title, are utilized.

Sptiz described duplicate document detection [6], Character Shape Coding information retrieval[7], shape based word recognition [8], and document reconstruction [9], without resorting to character recognition. Character shape codes encode whether or not the character in question fits between the baseline and the x-line or, if not, whether it has an ascender or descender , and the number and spatial distribution of the connected components. To get character shape codes, character cells must first be segmented. This method is therefore unsuitable for dealing with words with connected characters. Additionally, it is lexicon dependent, and its performance is somewhat affected by the appropriateness of the lexicon to the document being processed. In method described by Spitz and Tan character segmentation is a necessary step before downstream processes can be performed. In many cases, especially with document images of poor quality, it is not easy to separate connected characters. Moreover, the word, rather than the character, is normally a basic unit of useful meaning in document image retrieval.

A segmentation-free word image matching approach is used to avoid difficulties of touching characters, it treats each word object as a single, indivisible entity, and attempts to recognize it using features of the word as a whole.

1.2 Proposed Method

In this paper, an improved document image retrieval approach with the ability of matching partial word images has been proposed.

Step 1:. Word images are represented by feature strings.

Step 2: A feature string matching method based on dynamic programming is then utilized to evaluate the similarity between two feature strings or a part of them . The advantage of using the technique of inexact feature string matching, is that it can successfully handle not only kerning, but also words with heavily touching characters and it does not require any training prior to use. The proposed partial word image matching technique is somewhat similar to that of word recognition based on Hidden Markov Models.

The technique is tested in two applications of document image retrieval. One application is to search for user-specified words and their variations in document images (word spotting). The user's query word and a word image object extracted from documents are first represented by two respective feature strings. Then, the similarity between the two feature strings is evaluated. Based on the evaluation, we can estimate how the document word image is relevant to the user-specified word. For example, when the user keys in the query word "string." Our method detects words such as "strings" and "substring." Our experiments on real document images, including scanned books, student theses, journal/conference papers, etc., show promising performance by our proposed approach to word spotting in document images.

Fig. 1. Different spacing: (a) separated adjacent characters,(b) overlapped adjacent characters, and (c) touching adjacent characters

2.Description of feature string for the word image

It is presumed that word objects are extracted from document images via some image processing such as skew estimation and correction, connected component labeling, word bounding, etc. The feature employed in our approach to represent word bitmap images is the Left-to-Right Primitive String (LRPS), which is a code string sequenced from the leftmost of a word to its rightmost. Line and traversal features are used to extract the primitives of a word image. A word printed in documents can be in various sizes, fonts, and spacings. When we extract features from word bitmaps, we have to take this fact into consideration. Generally speaking, it is easy to find a way to cope with different sizes. For a word image in a printed text, two characters could be spaced apart by a few white columns caused by intercharacter spacing, as shown in Fig. 1a. But, it is also common for one character to overlap another by a few columns due to kerning, as shown in Fig. 1b. Worse still, as shown in Fig. 1c, two or more adjacent characters may touch each other due to condensed spacing. We are thus faced with the challenge of separating such touching characters. We shall utilize inexact feature string matching to handle the problem.

2.1 LRPS Feature Representation

A word is explicitly segmented, from the leftmost to the rightmost, into discrete entities. Each entity, called a primitive here, is represented using definite attributes. A primitive p is described using a two-tuple (σ,ω) where ρ is the Line-or-Traversal Attribute (LTA) of the primitive, and ω is the Ascender-and-Descender Attribute (ADA). As a result, the word image is expressed as a sequence P of pi's

$P=<p_1p_2\ldots p_n>=(\sigma_1, \omega_1)(\sigma_2, \omega_2)\ldots\ldots(\sigma_n, \omega_n)>$, (1)

where the ADA of a primitive $\omega \in \Omega$ = {"x," "a," "A," "D," "Q"}, which are defined as: D," "Q"}, which are defined as:

- "x": The primitive is between the x-line (an imaginary line at the x-height running parallel with the baseline, as shown in Fig. 2) and the baseline.
- "a": The primitive is between the top boundary and the x-line.
- "A": The primitive is between the top boundary and the baseline.
- "D": The primitive is between the x-line and the bottom boundary.
- "Q": The primitive is between the top-boundary and the bottom boundary

The definition of x-line, baseline, top boundary, and bottom boundary may be found in Fig. 2. A word bitmap extracted from a document image already contains the information of the baseline and x-line, which is a byproduct of the text line extraction from the previous stage.

2.2 Generating Line-or-Traversal Attribute

LTA generation consists of two steps. We extract the straight stroke line feature from the word bitmap first, as shown in Fig. 2a. Note that only the vertical stroke lines and diagonal stroke lines are extracted at this stage. Then, the traversal

features of the remainder part are computed. Finally, the features from the above two steps are aggregated to generate the LTAs of the corresponding primitives.

2.2.1 Straight Stroke Line Feature

A run-length-based method is utilized to extract straight stroke lines from word images. We use R (a,θ) to represent a

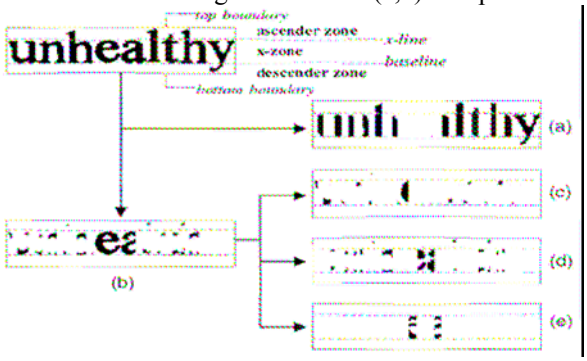

Fig. 2. Primitive string extraction (a) Straight stroke line features,(b) remainder part of (a) , (c) traversal T_N=2, (d) traversal T_N=4,(e) traversal T_N=6.

directional run, which is defined as a set of blackconcatenating pixels that contains pixel a, along the specified direction θ , | R (a,θ) | is the run length of R (a,θ), which is the number of black points in the run.

The straight stroke line detection algorithm is summarized as follows:.

1. Along the middle line (between the x-line and baseline), detect the boundary pair [A_l ,A_r] of each stroke, where A_l A_r are the left and right boundary points, respectively.
2. Detect the midpoint Am of a line segment A_l A_r
3. Calculate R (A_m,θ) for different θ s, from which We select θ max as the As's run direction.
4. If $|R(A_m,\theta_{max})|$ is near or larger than the x-height,the pixels containing Am, between the boundary points Al and Ar along the direction θ_{max}, are extracted as a stroke line.

In the example of Fig. 2, the stroke lines are extracted as in Fig. 2a, while the remainder is as in Fig. 2b.

According to its direction, a line is categorized as one of three basic stroke lines: vertical stroke line, leftdown diagonal stroke line, and right-down diagonal stroke line. According to the type of stroke lines, three basic primitives are generated from extracted stroke lines. Meanwhile, their ADAs are assigned categories based on their top-end and bottom-end positions. Their LTAs are respectively expressed as:

- "l": Vertical straight stroke line, such as in the characters "l," "d," "p," "q," "D," "P," etc For the primitive whose ADA is "x" or "D,"we further check whether there is a dot over the vertical stroke line. If the answer is "yes," the LTA of the primitive is reassigned as "i."
- "v": Right-down diagonal straight stroke line,such as in the characters "v," "w," "V," "W,"etc.
- "w": Left-down diagonal straight stroke line,such as in the characters "v," "w," "z," etc.For the primitive whose ADA is "x" or "A,"we further check whether there are two horizontal stroke lines connected with it at the top and bottom. If so, the LTA of the primitive is reassigned as "z."

2.2.2 Traversal Feature

After the primitives based on the stroke line features are extracted as described above, the primitives of the remainder part in the word image are computed based on traversal features. To extract traversal features, we scan the word image column by column, and the traversal number T_N is recorded by counting thenumber of transitions from black pixel to white pixel,or vice versa, along each column. This process is not carried out on the part represented by the stroke line features described above. According to the value of T_N, different feature codes are assigned as follows:

- "&": There is no image pixel in the column

It corresponds to the blank intercharacter space..

Fig 3.Different fonts

In case of kerning we can insert a space primitive between them.. If TN = 2, two parameters are utilized to assign it a feature code. One is the ratio of its black pixel number to xheight, κ. The other is its relative position with respect to the x-line and the baseline, $\varepsilon=D_m/D_b$, where D_m is the distance from the topmost stroke pixel in the column to the x-line and Db is the distance from the bottommost stroke pixel to the baseline.

- "n": , κ < 0:2 and ε < 0:3,
- "u": , κ < 0:2 and ε > 3, and
- "c": , κ > 0:5 and 0:5 < ε < 1:5.

If T_N> 4, the feature code is assigned as:

- "o": T_N = 4,
- "e": T_N = 6, and
- "g": T_N = 8.

Then, the same feature codes in consecutive columns are merged and represented by one primitive. Few columns may possibly have no resultant feature codes because they do not meet the requirements of the various eligible feature codes described above, which is usually the result of noise. Such columns are eliminated straightaway. As illustrated in Fig. 2, the word image is decomposed into one part with stroke lines (as in Fig. 2a) and other parts with different traversal numbers (Fig. 2c for T_N = 2, Fig. 2d for T_N =4, and Fig. 2e for T_N = 6). The primitive string is composed by concatenating the above generated primitives from the leftmost to the rightmost. The resultant primitive string of the word image in Fig.2 is:

(n,x)(l,x)(u,x)(o,x)(l,x)(u,x)(o,x)(l,x)(o,x)(n,x)(o,x)(l,x)
(u,x)(&,&)(o,A)(l,A)(o,x)(l,x)(n,x)(&,&)(c,x)(e,x)(o,x)
(&,&)(o,x)(e,x)(l,x)(u,x)(o,A)(l,A)(&,&)(n,x)(l,A)(o,x)o,A)
(l,A)(o,x)(n,x)(o,x)

2.3 Postprocessing

To be able to deal with different fonts, primitives should be independent of typefaces. Among various fonts, the significant difference that impacts the extraction of an LRPS is the presence of serif, specially at the parts expressed by traversal features.Fig. 3 gives some examples of where serif

is present and others, where it is not. It is a basic necessity to avoid the effect of serif in LRPS representation.primitives. For instance, a primitive "(u,x)" in a primitive subsequence "(l,x)(u,x)(&,&)" is normally Primitives produced by such serifs can thus be eliminated by analyzing its preceding and succeeding generated by a right-side serif of characters such as "a," "h," "m," "n," "u," etc. Therefore, we can remove the primitive "(u,x)" from a primitive subsequence "(l,x)(u,x)(&,&)." Similarly, a primitive "(o,x)" in a primitive subsequence "(n,x)(o,x)(l,x)" is normally generated by a serif of characters such as "h," "m," "n," etc. Therefore, we can directly eliminate the primitive "(o,x)" from a primitive subsequence "(n,x)(o,x)(l,x)."

With this postprocessing rules, the primitive string of the word in Fig. 2 becomes: (l,x)(u,x)(o,x)(l,x)(n,x)(l,x) (&,&)(l,A)(o,x)(l,x)(&,&)(c,x)(e,x)(o,x)(&,&)(o,x)(e,x) (l,x)(l,A)(&,&)(n,x)(l,A)(o,x)(l,A)(n,x)(l,x)(&,&)(y,D).

Based on the feature extraction described above, we can give each character a standard primitive string token (PST). For example, the primitive string token of the character "b" is "(l,A)(o,x)(c,x)," Table 1 lists the primitive string. tokens of all characters in the alphabet.

Ch	PST	Ch	PST
a	(o,x)(e,x)(l,x)	A	(w,A)(v,A)
b	(l,A)(o,x)(c,x)	B	(l,A)(e,A)(o,A)
c	(c,x)(o,x)	C	(c,A)(o,A)
d	(c,x)(o,x)(l,A)	D	(l,A)(o,A)(c,A)
e	(c,x)(e,x)(o,x)	E	(l,A)(e,A)
f	(n,x)(l,A)(u,a)	F	(l,A)(o,A)(u,a)
g	(g,D)(e,D)	G	(c,A)(o,A)(e,A)(o,A)
h	(l,A)(n,x)(l,x)	H	(l,A)(n,x)(l,A)
i	(i,A)	I	(l,A)
j	(i,Q)	J	(u,x)(l,A)
k	(k,x)	K	(k,A)
l	(l,A)	L	(l,A)(u,x)
m	(l,x)(n,x)(l,x)(n,x)(l,x)	M	(l,A)(v,A)(w,A)(l,A)
n	(l,x)(n,x)(l,x)	N	(l,A)(v,A)(l,A)
o	(c,x)(o,x)(c,x)	O	(c,A)(o,A)(c,A)
p	(l,D)(o,x)(c,x)	P	(l,A)(o,A)(c,A)
q	(c,x)(o,x)(l,D)	Q	(c,A)(o,A)(e,Q)(o,D)
r	(l,x)(n,x)	R	(l,A)(o,A)(e,A)(o,A)
s	(o,x)(e,x)(o,x)	S	(o,A)(e,A)(o,A)
t	(n,x)(l,A)(o,x)	T	(u,a)(l,A)(u,a)
u	(l,x)(u,x)(l,x)	U	(l,A)(u,x)(l,A)
v	(v,x)(w,x)	V	(v,A)(w,A)
w	(v,x)(w,x)(v,x)(w,x)	W	(v,A)(w,A)(v,A)(w,A)
x	(x,x)	X	(x,A)
y	(y,D)	Y	(Y,A)
z	(z,x)	Z	(z,A)

TABLE 1. Primitive String Tokens of Characters

3 Inexact Feature Matching

Each word image is described by a primitive string.The word searching problem can then be stated as finding a particular sequence/subsequence in the primitive string of a word. The procedure of matching word images then becomes a measurement of the similarity between the string A=[a1;a2;….an] representing the features of the query word and the string B=[b1; b2; . . . ; bm] representing the features of a word image extracted from a document. Matching partial words becomes evaluating the similarity between the feature string A with a subsequence of the feature string B. For example, the problem of matching the word "health" with the word "unhealthy" in Fig. 2 is to find whether there exists the subsequence A=(l,A)(n,x)(l,x)(&,&)(c,x)(e,x)(o,x)(&,&)(o,x)(e,x)(l,x)(&,&)(l,A)(&,&)(n,x)(l,A)(o,x)(&,&)(l,A)(n,x)(l,x) in the primitive sequence of the word "unhealthy."

We apply the technique of inexact string matching to evaluate the similarity between two primitive strings, one from a template image and the other from a word image extracted from a document. For a primitive string A of length n and a primitive string B of length m, V(i,j) is defined as the similarity value of the prefixes [a1; a2; . . . ; ai] and [b1; b2; . . . ; bj]. The similarity of A and B is precisely the value V(n,m).

The similarity of two strings A and B can be computed by dynamic programming with recurrences .

$$\forall i,j: \quad \begin{cases} V(i,0) = 0 \\ V(0,j) = 0. \end{cases} \qquad (2)$$

The general recurrence relation is: For

$$1 \le i \le n, 1 \le j \le m:$$

$$V(i,j) = max \begin{cases} 0 \\ V(i-1,j-1) + \epsilon(a_i,b_j) \\ V(i-1,j) + \mu(a_i,-) \\ V(i,j-1) + \nu(-,b_j). \end{cases} \qquad (3)$$

In our experiments, the score of matching any primitive with the the space primitive "–" is defined as: $\mu(ak,\text{-}) = \upsilon(\text{-},bk) = -1$ *for* $ak \neq \text{-}\ bk \neq \text{-}$ (4)

$\mu(ak, \&) = \upsilon(\&, bk) = -1$ *for* $ak \neq \&\text{-}bk \neq \&\text{-}$, (5)

and the matching score between two primitives "&"is given by

ε (&,&)=2 (6)

while the matching score between two primitives ai and bj is defined as ε (ai,bj)= ε ((σai ,ωai),(σb j,ωbj)= 1(σai,, σb j) + ε2 (ωai,, ωbj) (7)

where ε1 is the function specifying the match value between two elements x' and y' of LTA. It is defined as:

ε1(x',y') = {1 if x'=y' (8)

(-1 else

Similarly, ε2 is the function specifying the match value between two elements x" and y" of ADA. It is defined as:

ε2(x", y")= {1 if x"=y" (9)

{-1 else

Finally, maximum score is normalized to the interval [0,1], with 1 corresponding to a perfect match:

$$S_1 = \max_{\forall i,j} V(i,j)/V_A^*(n,n), \qquad (10)$$

where VA *(n,n)matching score between the string A and itself. The maximum operation in (10) and the restarting recurrence operation in (2) ensure the ability of partial word matching. If the normalized maximum score in (10) is greater than a predefined threshold δ, then we recognize that one word image is matched with the other (or portion of it).

On the other hand, the similarity of matching two whole word images in their entirety (i.e., no partial matching is allowed) can be represented by: The problem can be evaluated systematically using tabular computation. In this approach, a bottom-up approach is used to compute *V(i,j)*We first compute *V(i,j)* for the smallest possible values of i and j, and then compute the value of *V(i,j*)for increasing values of i and j. Table 2 illustrates the score table computing

$$S_2 = V(n,m)/min(V_A^*(n,n), V_B^*(m,m)). \qquad (11)$$

Table 2 Scoring table for computing *V(i,j)*

by aggregating the characters' primitive string tokens according to the character sequence of the word, with the special primitive "&" inserted between two adjacent PSTs. It can be seen that the maximum score achieved in the table corresponds to the match of the character sequence "health" in the word "unhealthy.

4 Application : Word Searching

We now focus on the application of our proposed partial word image matching method to word spotting. Searching/locating a user-specified keyword in image format documents has been a topic of interest for many years. It has its practical value for document information retrieval. For example, by using this technique, the user can locate a specified word in document images without any prior need for the images to be OCR-processed.

4.1 System Overview

The overall system structure is illustrated in Fig. 4.When a document image is presented to the system, it goes through preprocessing, as in many document image processing systems. It is presumed that processes such as skew estimation and correction if applicable, and other image-quality related processing, are performed in the first module of the system.

Then, all of the connected components in the image are calculated using an eight-connected component analysis algorithm. The representation of each connected component includes the coordinates and dimensions of the bounding box and so on.Word objects are bounded based on a merger operation on the connected components. As a result, the left, top, right, and bottom coordinates of each word bitmap are obtained. Meanwhile, the baseline and x-line locations in each word are also available for subsequent processing. Extracted word bitmaps with baseline and x-line information are the basic units for the downstream process of word matching, and are represented with the use of primitive strings as
described in Section 2.

When a user keys in a query word, the system generates its corresponding word primitive token (WPT) by aggregating the characters' primitive string tokens according to the character sequence of the word, with the special primitive "&" inserted between two adjacent PSTs. For example, the WPT of the word "health"is:
(l,A)(n,x)(l,x)(&,&)(c,x)(e,x)(o,x)(&,&)(o,x)(e,x)(l,x)
(&,&)(l,A)(&,&)(n,x)(l,A)(o,x)(&,&) (l,A)(n,x)(l,x).

This primitive string is matched with the primitive string of each word object in the document image to measure the similarity between the two. According to the similarity measurement, we can estimate how the word in the document image is relevant to the query word.

4.2 Experimental Results

Experiment is carried out on the digital library of our university stores which has a huge number of old student theses and books deemed to make these documents accessible on the Internet. NUSDL document image database is generated using PDF files
of scanned books and students theses.

In Fig. 6. The document image is stored in a PDF file which is accessible at the Web site of IEEE Xplore online publications. The words "string," "strings," and "substring" in different fonts are correctly detected and located in the document when "string" is used as the query word.

5 Conclusion

Document images have become a popular information source in our modern society, and information retrieval in document image databases is an important topic in knowledge and data engineering research. Document image retrieval without OCR has its practical value, but it is also a challenging problem. In

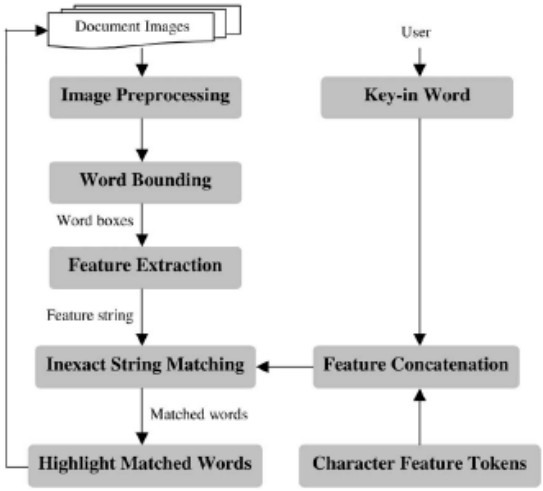

Fig 5. System diagram for searching user specified word in document image

knowledge and data engineering research. Document image retrieval without OCR has its practical value, but it is also a challenging problem. Inthis paper, we have proposed a word image matching approach with the ability of matching partial words. To measure the similarity of two word images , each word image is represented by a primitive string. Then, an inexact string matching technique is utilized to measure the similarity between the two primitive strings generated from the two word images. Based on the matching score, we can estimate how one word image is relevant to the other and decide whether one is a portion of the other. The proposed word image approach has been applied to two document image retrieval problems, i.e., word spotting and document similarity measurement. Experiments have been carried out on various document images such as scanned books, students theses, and journal/conference papers downloaded from the Internet, as well as UW document images. The test results confirm the validity and feasibility of the applications of the proposed word matching approach to document image retrieval.

REFERENCES

[1] D. Doermann, "The Indexing and Retrieval of Document Images:A Survey," *Computer Vision and Image Understanding,* vol.70, no. 3,pp. 287-298, 1998.
[2] M. Mitra and B.B. Chaudhuri, "Information Retrieval fromDocuments: A Survey," *Information Retrieval*, vol. 2, nos.2/3,pp. 141-163, 2000.
[3] D. Niyogi and S. Srihari, "The Use of Document Structure Analysis to Retrieve Information from Documents in Digital Libraries*," Proc. SPIE, Document Recognition IV,* vol. 3027, pp.207-218, 1997.
[4] Y.Y. Tang, C.D. Yan, and C.Y. Suen, "Document Processing for Automatic Knowledge Acquisition," *IEEE Trans. Knowledge and Data Eng*., vol. 6, no. 1, pp. 3-21, Feb. 1994.
[5] Y. He, Z. Jiang, B. Liu, and H. Zhao, "Content-Based Indexing and Retrieval Method of Chinese Document Images*," Prof. Fifth Int'l Conf. Document Analysis and Recognition (ICDAR '99),* pp.685-688,1999.
[6] A.L. Spitz, "Duplicate Document Detection," *Proc. SPIE,Document Recognition IV*, vol. 3027, pp. 88-94, 1997.
[7] A.F. Smeaton and A.L. Spitz, "Using Character Shape Coding for Information Retrieval," *Proc. Fourth Int'l Conf. Document Analysis and Recognition,* pp. 974-978, 1997.
[8] A.L. Spitz, "Shape-Based Word Recognition," *Int'l J. Document Analysis and Recognition,* vol. 1, no. 4, pp. 178-190, 1999.
[9] A.L. Spitz, "Progress in Document Reconstruction," *Proc. 16th Int'l Conf. Pattern Recognition,* vol. 1, pp. 464-467, 2002.
[10] Z. Yu and C.L. Tan, "Image-Based Document Vectors for Text Retrieval," *Proc. 15th Int'l Conf. Pattern Recognition*, vol. 4, pp.
393- 396, 2000.