Document Image Recognition Based on Template Matching of Component Block Projections

Hanchuan Peng, *Member*, *IEEE*, Fuhui Long, and Zheru Chi, *Member*, *IEEE*

Abstract—Document Image Recognition (DIR), a very useful technique in office automation and digital library applications, is to find the most similar template for any input document image in a prestored template document image data set. Existing methods use both local features and global layout information. In this paper, we propose a novel algorithm based on the global matching of Component Block Projections (CBP), which are the concatenated directional projection vectors of the component blocks of a document image. Compared to those existing methods, CBP-based template-matching methods possess two major advantages: 1) The spatial relationship among the component blocks of a document image is better represented, hence a very high matching accuracy can be obtained even for a large template set and seriously distorted input images; and 2) the effective matching distance of each template and the triangle inequality are proposed to significantly reduce the computational cost. Our experimental results confirm these advantages and show that the CBP-based template-matching methods are very suitable for DIR applications.

Index Terms—Document image recognition, template matching, component block projection.

1 Introduction

NOWADAYS, a large amount of existing paper documents are transformed to digital document images through scanners or cameras. Efficient storage, retrieval, and management of these document image archives are extremely important in many office automation and digital library applications. As a result, techniques for automatic document image analysis are highly demanded. A typical framework for a document image analysis system is given in Fig. 1, where Document Image Recognition (DIR) is to recognize the type of an input document image (or "query image"). Given that document types are defined via the prestored document template images (which can either be stored as physical images or described with a language such as XML), DIR is often implemented as finding the most similar template for an input document image. Obviously, a fast and accurate DIR algorithm will be very helpful for the consequent automatic registration, annotation, and text recognition of document images.

Existing DIR techniques [3], [5], [6], [9], [10], [11], [12], [14], [15], [17], [18], [19], [20] can be roughly divided into two categories. Methods of the first category rely on matching local features. For instance, Lopresti [11] used the approximate string matching of recognized characters for document recognition; Tseng and Chen [19] registered forms based on three types of line segments; Fan and Chang [6] registered forms using a line crossing relationship matrix; Cesarini et al.'s form-reader system [3] used attributed relational graphs; Shimotsuji and Asano [18] presented a 2D hash-table cell structure to identify different forms; Watanabe et al. [20] described blank form structures with the repetitions and positions of cells; Safari et al. [17] proposed a projective geometry method to map an input document to a template document. Many methods based on matching local features are sensitive to distortions of document images and the misdetection of local features (which are common in general document images), and are often limited to particular types of documents. The second category of DIR techniques combines both local features and global layout information. For instance, Hull [10] imposed a grid to the CCITT G4 pass-code maps of document images and consequently composed feature vectors for recognition. Hu et al. [9] proposed interval code to describe the spatial layout of document images; Peng et al. [14], [15] used Component-Block-List (CBL) matching to recognize document images with general layout and contents. Compared to the first category, methods in the second category often produce relatively better recognition accuracy, although they are still subject to great improvement for real applications.

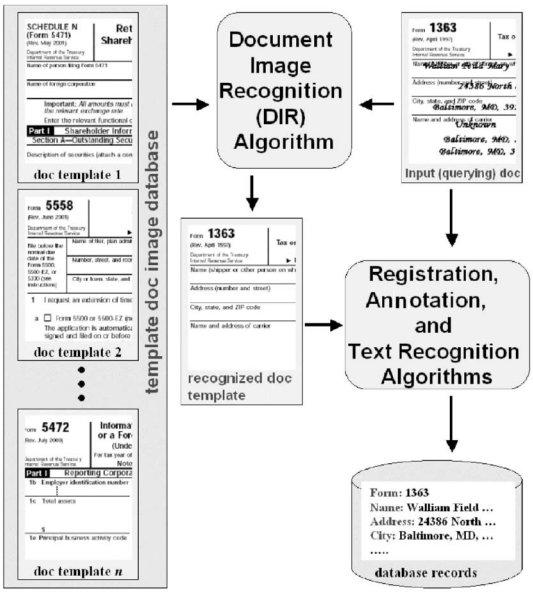


Fig. 1. Block diagram of a typical document image analysis system.

In this paper, we propose a superior DIR algorithm which can find many real applications such as automatic form data reading, document sharing in video-conferencing systems, document image retrieval, etc. The new method uses the directional projections of component blocks of document images to produce very high-recognition accuracy for images with large deformations. This paper is organized as follows: Section 2 presents our approach in detail. Section 3 shows four sets of experiment results. Section 4 gives a brief discussion and conclusion.

2 Methodology

Our goal is to develop an *accurate* and *computationally efficient* method for DIR applications. Both requirements are very important because of the great impact of the DIR performance on subsequent procedures in a large document image processing system (Fig. 1). Here, we present a global matching algorithm using directional component block projections.

2.1 Component Block Projection Vectors

We produce the component blocks of a document image using a document image processing package, PageX [13]. A scanned gray-scale document image shown in Fig. 2a (which consists of heterogeneous contents) is binarized and rotated to the upright

- *H. Peng is with the NERSC Division, Lawrence Berkeley National Laboratory, One Cyclotron Road, MS 50F, Berkeley, CA 94720 and is also with the Center for Biomedical Image Computing, Department of Radiology, Johns Hopkins University, School of Medicine, Baltimore, MD 21287 E-mail: hpeng@lbl.gov.*
- *F. Long is with Duke University Medical Center, Box 3209, Durham, NC 27710. E-mail: long@neuro.duke.edu.*
- *Z. Chi is with the Center for Multimedia Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong. E-mail: enzheru@polyu.edu.hk.*

Manuscript received 3 Jan. 2002; revised 25 July 2002; accepted 23 Dec. 2002. Recommended for acceptance by M. Pietikainen.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 115640.

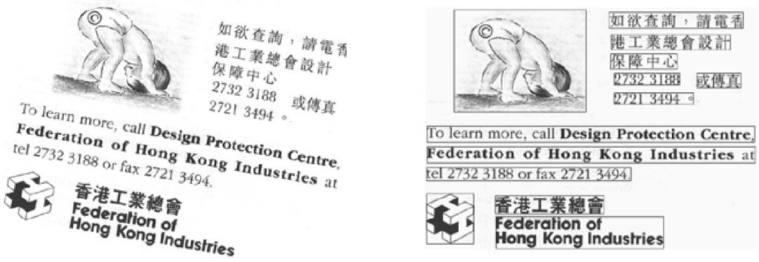


Fig. 2. Component Block Representation (CBR) of a document image. (a) The scanned grayscale image. (b) The CBR of (a) (component blocks are drawn as the rectangular bounding boxes of the corresponding document image regions).

position as shown in Fig. 2b; then the component blocks, i.e., the rectangular bounding boxes of the isolated content regions (i.e., texts or graphics) are extracted (Fig. 2b). (See [14], [15] for the details of image preprocessing.) The width of component block edges is defined as one pixel. We call the union of all component blocks in Fig. 2b the "Component Block Representation" (CBR) of the document image shown in Fig. 2a. Without losing any generality, CBR can be viewed as a binary image of rectangular boxes, where foreground pixels (i.e., box edges) take value 1 and background pixels take value 0. We can write a CBR as an array

$$\begin{bmatrix} b_{11} & \cdots & b_{1w} \\ \vdots & b_{nm} & \vdots \\ b_{h1} & \cdots & b_{hw} \end{bmatrix}$$

$(1 \le m \le w; 1 \le n \le h)$, where w and h are the image width and height, respectively; $b_{nm} = 1$ if the pixel $\{n, m\}$ is on the block edge, otherwise $b_{nm} = 0$. Since CBR is independent of the concrete block contents of a document, they can be used to deal with documents of general contents (i.e., texts including paragraphs, sentences or words in different languages and fonts, and graphics including images, drawings, or logos) [14], [15].

To recognize an input document image as one of the prestored templates, we obtain CBRs of both the input and template images and find the best matching pair. Rather than using blocks as features [14], [15], we define the directional projection of each component block as the sum of pixel values across the respective CBR image region in the specific direction. The projection vector of all blocks together (in the sense of union) is called the global projection. Hence, the global horizontal and vertical projection vectors are $[\Sigma_{m=1}^{w} b_{1m}, \cdots, \Sigma_{m=1}^{w} b_{hm}]$ and $[\Sigma_{n=1}^{h} b_{n1}, \cdots, \Sigma_{n=1}^{h} b_{nw}]$, respectively. We concatenate them (first horizontal and then vertical) as a long feature vector with $L = (w + h)$ bins. Since each bin can be regarded as one dimension, a document image is represented as a point in the L-dimensional space in term of the global Component Block Projection (CBP) vector.

CBP vectors have several valuable properties. First, they reflect the spatial relationships of component blocks and the local variations of individual component blocks. The changes of the relative positions of blocks will lead to variations of either the horizontal projection vector or the vertical projection vector or both. Second, the global projection vector is robust to local block variations, that is, individual block variations will only cause localized variation to the global projection vector. Third, CBP vectors allow the following useful interpretation of block deformations. The variation of the global horizontalprojection vector can be expressed as the sum over the variations of one-pixel-wide columns (i.e., vertical lines) of the binary CBR image. The variation of the global vertical projection vector is the sum over the variations of one-pixel-thick rows (i.e., horizontal lines) of the binary CBR image. Note that the generality of document contents indicates these variations can appear anywhere. Therefore, we can assume that the local variation of each row or column has an independent identical distribution and we can approximate the global projection variation with the Gaussian distribution based on the central limit theorem:

$$\begin{aligned} p(\Delta g(T)) &= p(g(Q) - g(T)) \\ &= \frac{1}{\beta} \exp\{-\alpha [g(Q) - g(T)]^2\} = p(g(Q)|g(T)), \end{aligned} \tag{1}$$

where $g(Q)$ and $g(T)$ are the concatenated global projection vectors of the input CBR Q and the corresponding template CBR T, respectively, $\Delta g(T)$ is the variation of $g(T)$, α and β are two positive parameters. The last equal mark is valid because for any specific T, $(g(Q)|g(T))$, and $(\Delta g(T))$ are the same event. Equation (1) indicates that, as long as the number of component blocks is large enough, the distribution of the global projection variations will not depend on any specific distribution of local block variations. Therefore, it is possible to find a general solution to the DIR problem without an analysis of more complicated local deformations. According to (1), finding the optimal template T for an input CBR Q is equivalent to maximizing $p(\Delta g(T))$, which further equals maximizing the Gaussian function in (1).

2.2 Global Matching Methods

Given a template CBR set $S(T) = \{T_i, i = 1, 2, \ldots, K\}$, where K is the total number of templates, i.e., $|S(T)|$, for an input CBR Q, we find the template CBR T^* with the maximum posterior probability $p(T_i|Q)$, i.e.,

$$T^* = \arg\max_{T_i} p(T_i|Q).$$

With the Bayesian theorem, we have

$$T^* = \arg\max_{T_i} \frac{p(T_i)p(Q|T_i)}{p(Q)} = \arg\max_{T_i} p(T_i)p(Q|T_i).$$

For most applications where there is no priority of individual templates, the prior distribution $p(T_i)$ can be set as uniform and we have the maximum-likelihood scheme, i.e.,

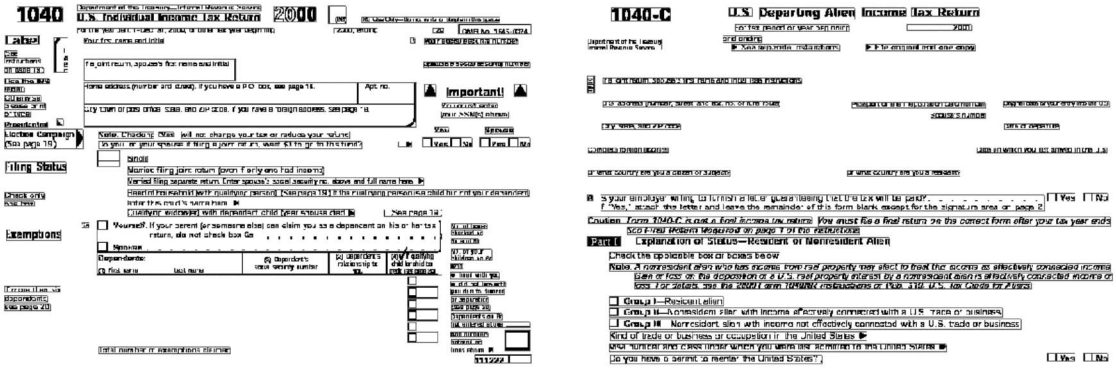
$$T^* = \arg\max_{T_i} p(Q|T_i).$$

Based on global projection vectors, template T^* can be obtained as follows:

$$T^* = \arg\max_{T_i} p(g(Q)|g(T_i)). \tag{2}$$

Considering (1), we find T that is closest to Q.The following Naive-Global matching method ("Naive-Global" for short) is used to get the best T^*:

$$T^*(Q) = \arg\min_{T \in S} D(g(Q), g(T)), \tag{3}$$



(a) (b)

Fig. 3. CBR examples in the data set USTAX208. (a) Form *1040*. (b) Form *1040C*.

where $D(.)$ is the distance metric for the pair-wise CBP vectors $\{g(Q), g(T)\}$. Among many possible choices for $D(.)$ [16], here we choose L_1 distance due to its computational simplicity and proportional weighting of the difference.

Since Naive-Global compares Q with every template in $S(T)$, the computational complexity is $O(KL) = O(K(w+h))$, where L is the length of the concatenated global projection vector, w and h are the document image width and height, respectively.

We want to substantially reduce the complexity of Naive-Global without sacrificing the recognition accuracy. Rather than considering multidimensional indexing techniques (for example, see [1], [8]), we propose alternate methods as follows.

We define the Effective Matching Distance (EMD) of template T_i as half of the minimum distance between pair-wise templates T_i and T_j, i.e.,

$$E(T_i) = \frac{1}{2} \min_{j \in [1,K], j \neq i} D(T_i, T_j).$$

Clearly, if an input CBR Q has a distance $D(Q, T_i) < E(T_i)$, we can immediately conclude that T_i is the best matching template. If this condition is not satisfied for all templates in $S(T)$, we need to compare $D(Q, T_i)$ over all T_is to find the minimum value. As EMDs of all templates can be precalculated offline and will not affect the matching time, theoretically the average computational complexity of this method is $\mathrm{O}(KL/2)$. We call this EMD-based method "Efficient-Global matching" ("E-Global" for short) because it can double the speed of Naive-Global.

The efficiency of E-Global relies on the assumption that the deformation is not too large, i.e., suppose Q is a deformation of T, we expect $D(Q, T)$ is smaller than the corresponding $E(T)$. We call this assumption "weak-deformation." This assumption is actually true for many real applications. However, once the deformation is large so that $D(Q, T) \geq E(T)$, E-Global will automatically reduce to Naive-Global. Therefore, the practical computational complexity of E-Global is usually between $\mathrm{O}(KL/2)$ and $\mathrm{O}(KL)$, depending on the deformation degree of Q with respect to the corresponding T.

With the weak-deformation assumption, we can further reduce the computation using the triangle inequality. Taking two templates T_i, T_j, and the input Q as three points in the high-dimensional space, we have two triangle inequalities $D(Q, T_i) + D(T_i, T_j) \geq D(Q, T_j)$ and $D(Q, T_i) + D(Q, T_j) \geq D(T_i, T_j)$. If the condition

$$\left| D(Q, T_j) - D(T_i, T_j) \right| \geq E(T_i) \tag{4}$$

is satisfied, we can derive $D(Q, T_i) \geq E(T_i)$, i.e., the distance between T_i and Q will be no less than the EMD of T_i. As mentioned above, under the weak deformation assumption, the matched template T of input Q should satisfy $D(Q, T) < E(T)$. Therefore, if (4) holds, T_i can be excluded immediately from being a matched template of Q. In this way, the L subtractions in the distance calculation are reduced to one subtraction in (4). Next, among all the templates for which (4) is not satisfied, we can use E-Global to find out the optimal matching template. We call this method "Turbo-Global matching" ("T-Global" for short) because it uses the triangle inequality to accelerate E-Global. Since all $D(T_i, T_j)$ and $E(T_i)$ are computed beforehand, we only need to compute one distance $D(Q, T_j)$ (say, $j = 1$) and check whether (4) holds for each T_i. In practice, the computational complexity of T-Global is between $\mathrm{O}(K + L - 1)$ and $\mathrm{O}(KL)$. Independently, similar usage of the triangular inequality was also noticed in other applications [2], [4], [7].

In summary, T-Global is the fastest method; E-Global comes next; Naive-Global is slowest. They have the same recognition accuracy. We call them CBP-matching methods hereafter.

3 Experimental Results

We tested the accuracy (in term of the recognition rate r_c, i.e., the percentage of document images that are correctly recognized) and the efficiency of our new methods. We also compared our methods with Hull's pass-code method [10] (called "PC-matching" here), Hu et al.'s interval-code method [9] (called "IC-matching"), and Peng et al.'s method [14], [15] (called "CBL-matching"). For the results reported below, the parameters for different methods tested were chosen according to the respective papers [9], [10], [15], i.e., a 4×4 grid for PC-matching and a 50×30 grid for IC-matching (we tried many other parameters but got similar or worse results). The parameters for CBL-matching are explained later.

3.1 Data Sets

We used two large-scale data sets, i.e., P1000 and USTAX208. We generated P1000 by excluding 350 very similar templates in an earlier database of 1,350 document templates reported in [14], [15]. Together with P1000, there is a deformation-generating program that produces test (query) CBRs with simulated deformations. This program simulates various image deformations caused by filled-in document contents, noise, and blocking errors [14], [15]. The parameters that can be set in this program include the block misdetection rate P_m, the block misaddition rate P_a, the block size deformation rate P_s, the size deformation scale factor S_s, the block location displacement rate P_d, the displacement scale factor S_d, the rotation probability P_r, and the rotation angle D_r.

The second data set, USTAX208, contains 208 templates of tax forms of the US Internal Revenue Service (IRS) (http://www.irs.gov/formspubs/index.html) and 1,040 test forms filled with different pseudotaxpayer information (the size of each image is 800×600 pixels). Two sample CBRs of the IRS forms *1040* and *1040C* are shown in Fig. 3. As a thorough analysis of the local

TABLE 1
Relationship between r_c (%) and $|S(T)|$

$\|S(T)\|$	CBL	IC	CBP
50	96.40	61.00	99.77
100	96.02	62.75	99.72
200	94.19	60.25	99.59
500	92.50	59.50	99.41
1000	91.13	57.90	99.25

differences among these forms is time-consuming, inaccurate, and subject to changes (e.g., due to the annually updated form columns), many local feature-based methods are not applicable because distinctive local features are not carefully designed in these forms. The importance of blocks in a CBR is indexed in terms of their areas. In the experiments, we only used the N_{CBR} (a presumed number) largest blocks. (More details of both data sets are available on request.)

3.2 Recognition Accuracy versus Template Set Size

We used P1000 to compare the recognition accuracy of CBP, CBL, and IC-matching with respect to different sizes of the template set, i.e., $|S(T)|$. Without a priori knowledge, from P1000 we arbitrarily selected four nonoverlapping subsets with 50, 100, 200, and 500 templates, respectively. As used in [14], [15], the parameters of the deformation function were set as $\{P_m = 0.2, P_a = 0.2, P_s = 0.2, S_s = 0.2, P_d = 0.5, S_d = 0.5, P_r = 0.5, D_r = 15^\circ\}$ (called DPC1 hereafter). They resulted in significantly deformed test CBRs [14], [15], many of which have larger deformations than real cases (as shown in Section 3.4). In each trial of the experiment, we generated 20,000 test CBRs (10 times the quantity in [15]).

Table 1 shows results of the recognition rate, r_c, with respect to $|S(T)|$. r_c of CBP-matching is always larger than 99 percent and is not sensitive to $|S(T)|$; evidently CBP-matching is better than CBL and IC-matching. Of note a larger template set often allows higher chance for misrecognize a test image, unless the features are very distinctive. This suggests CBP vectors are much more distinctive than CBL features and the interval code. The results shown in Table 1 also indicate CBP-matching is scalable to large DIR problems.

3.3 Recognition Accuracy versus CBR Block Number

We used the data set USTAX208 to compare the recognition accuracy of CBP, CBL, PC, and IC-matching, with respect to different numbers of CBR blocks, i.e., N_{CBR}. For each of the 208 template forms and the 1,040 test forms, we adjusted N_{CBR} from 10 to 60. A larger N_{CBR} indicates a more similar CBR to the respective form image.

Table 2 shows r_c of different methods with respect to N_{CBR}. Obviously, CBP-matching produces the highest r_c, which is significantly better than the other three methods. r_c of CBP-matching reaches the maximum when the 20 largest CBR blocks are used and does not degrade when more blocks are added. This indicates that CBP-matching makes a proper use of the most important features and performs robustly to additional information. In contrast, for CBL, PC, and IC-matching methods, the respective r_c reaches the highest value when about the 20 largest blocks are used, but degrades with additional blocks. This sensitivity to N_{CBR} is a result of the local matching error of every block in CBL-matching [14], [15], every row in IC-matching [9], or every bin in PC-matching [10]. As we neither expect more features would lower the performance nor want to iterate the matching procedure (which will increase the computational cost) to find the optimal N_{CBR}, this sensitivity is a critical disadvantage.

TABLE 2
Relationship between r_c (%) and N_{CBR}

N_{CBR}	CBL	PC	IC	CBP
10	77.78	91.34	86.54	99.60
20	88.18	95.19	92.02	99.80
30	87.58	94.90	92.40	99.80
40	85.56	94.32	91.44	99.80
50	81.72	93.26	90.67	99.80
60	81.52	92.88	87.12	99.90

TABLE 3
Deformation Characteristics Table (DeCT)

N_{CBR}	DeCT (CBL)				DeCT (CBP)			
	DPC1	DPC2	DPC3	DPC4	DPC1	DPC2	DPC3	DPC4
10	27.11	56.34	62.50	85.48	32.40	87.98	97.88	100.0
20	32.78	62.88	71.82	87.30	32.88	95.76	99.61	100.0
30	39.23	73.07	80.86	94.51	35.38	98.26	99.90	100.0
40	49.71	81.44	88.46	96.25	42.12	99.03	100.0	100.0
50	56.53	85.48	88.84	95.19	45.23	99.51	100.0	100.0
60	61.44	85.48	88.07	91.44	50.87	99.23	100.0	100.0

The better performance of PC-matching over both CBL and IC-matching is expected because it produces feature vectors by summing up local pass code on grid-cells and thus can be understood in a way similar to CBP-matching. However, since it still uses block contents (i.e., texts) to generate feature vectors, it can hardly outperform CBP-matching when there are a lot of variations in document contents.

3.4 A Comparison between Simulated Deformations and Real Deformations

One important question to ask is how well the deformations produced by the deformation-generating program of P1000 in Section 3.2 replicate the deformations of real data (e.g., USTAX208). To answer this question, we adjusted parameters of the deformation-generating program to create a set of simulated test CBRs using real templates, varying from strong deformations to weak deformations. We observed the range of parameters within which the simulated deformations were closest to the deformations of real data. Obviously, a DIR method obtains better recognition accuracy when the data have weaker deformations, which implies that we can compare the deformation degrees via the various recognition accuracies obtained. Thus, for each configuration of the deformation parameters, we depicted columns of r_c results like those of $r_c(\mathrm{CBL})$ and $r_c(\mathrm{CBP})$ in Table 2. The generated table is called Deformation Characteristics Table (DeCT) of a Deformation Parameter Configuration (DPC). The deformation-generating program in P1000 was applied to the templates in data set USTAX208 to examine whether the simulated deformations were comparable to the real cases. We defined four gradually weakening deformations:

1. **DPC1**: {as used in Section 3.2};
2. **DPC2**: $\{P_m = 0.10, P_a = 0.10, P_s = 0.10, S_s = 0.10, P_d = 0.3, S_d = 0.3, P_r = 0.3, D_r = 15^\circ\}$;
3. **DPC3**: $\{P_m = 0.10, P_a = 0.10, P_s = 0.10, S_s = 0.10, P_d = 0.2, S_d = 0.2, P_r = 0.2, D_r = 10^\circ\}$;
4. **DPC4**: $\{P_m = 0.05, P_a = 0.05, P_s = 0.05, S_s = 0.05, P_d = 0.1, S_d = 0.1, P_r = 0.1, D_r = 10^\circ\}$.

The obtained DeCTs are illustrated in Table 3. Comparing the $\mathrm{DeCT(CBP)}$ and $\mathrm{DeCT(CBL)}$ results with the respective $r_c(\mathrm{CBP})$ and $r_c(\mathrm{CBL})$ in Table 2, we see that the real deformation due to filled-in contents is comparable to the simulated deformations generated with DPC3 (and, for CBP-matching, it is even weaker, in the range of DPC3 and DPC4) and much weaker than those simulated deformations generated with DPC1 (as used in Section 3.2). Hence, we conclude that 1) the deformation-generating program in P1000 can be used to assist the study of DIR algorithms, generating meaningful results as in Section 3.2 and 2) the weak deformation assumption in Section 2.2 appears reasonable for the USTAX208 data set.

3.5 Computational Efficiency

Here, we investigated the computational efficiency of E-Global and T-Global. We define R_E and R_T as the percentages of test images to

TABLE 4
Relationship of the Computational Cost, $|S(T)|$, and N_{CBR}

$\|S(T)\|$	R_E (%)				R_T (%)			
	DPC1	DPC2	DPC3	DPC4	DPC1	DPC2	DPC3	DPC4
50	0	8.10	41.75	96.45	0.88	0.18	0.16	0.01
100	0	8.35	40.00	95.85	1.74	0.29	0.29	0.02
200	0	6.45	36.20	95.65	3.89	0.79	0.74	0.09
500	0	6.85	36.15	95.50	9.87	2.05	1.74	0.20
1000	0	6.85	36.70	95.30	18.88	3.35	3.30	0.47

N_{CBR}	R_E (%)	t_E (s)	R_E (%)	t_T (s)
10	94.55	11.69	5.48	10.94
20	98.28	11.34	6.32	11.11
30	99.09	11.77	6.22	11.09
40	99.09	11.08	6.10	11.08
50	98.99	11.69	5.90	10.90
60	99.09	11.23	6.08	11.01

R_E and R_T: Percentages of templates that satisfy the respective conditions for applying E-Global and T-Global; t_E and t_T: The total time cost by E-Global and T-Global to recognize the 1,040 real test images.

which E-Global and T-Global can be applied, respectively. They are actually ratios of test images that satisfy the weak-deformation assumption and (4), respectively.

For P1000, we first considered R_E with respect to various deformations of the test CBRs (generated by the parameter configurations DPC1 to DPC4) and the size of the template set, i.e., |S(T)|. As shown in Table 4, R_E increases significantly as the deformation decreases from DPC1 to DPC4. This indicates that the theoretical computational complexity of E-Global, i.e., about half of the computational load of Naive-Global, can be attained for test images with the deformation strength like DPC4. On the contrary, R_T decreases as the deformation decreases from DPC1 to DPC4, mainly because templates in P1000 seldom satisfy (4); however, this side effect appears to be minor because the computational complexity for weak deformations is largely governed by E-Global. In addition, R_T increases with $|S(T)|$; this is mainly because the volume of the projection vector space is limited, a larger $|S(T)|$ makes the projection vector space become denser and each template have a smaller EMD, therefore, (4) is more easily satisfied.

For USTAX208, with our implementation in Matlab/C++ on a PC with PIII 1GHz CPU running Linux 7.0, Naive-Global takes about 21 seconds (total time) to recognize 1,040 real test images, while E-Global and T-Global only take about 11 seconds, as shown in the t_E and t_T columns of Table 4. A computation reduction of near 50 percent is in good accordance with our analysis in Section 2.2. Table 4 also shows that R_E is large (> 94 percent) for all N_{CBR}, indicating most test forms have weak deformations; R_T is around 6 percent, which anyhow still contributes to a speed improvement.

4 Discussion and Conclusion

The experimental results indicate that our CBP-matching (i.e., Naive-Global, E-Global, and T-Global) methods are scalable to real DIR applications. These methods demonstrate high recognition accuracy and computational efficiency for various types of document images (simulated or real), various sizes of template sets, and various document image deformations. The good performance of our approach attributes to the following factors: 1) A better representation of the spatial relationship of component blocks: The concatenated global directional projection vectors have three advantages in representing document images. First, the projection vector well represents the spatial relationships of component blocks. Second, they allow canonical and economical computation. Third, they can be regarded as the sum of pixel-level deformations, which facilitates effective computation in a probabilistic framework. 2) A good mathematical approximation of block deformations: Although the Gaussian approximation itself has been applied to a variety of applications, as far as we know, this paper is the first attempt to apply it to DIR problems. This approach allows using the intuitive matching methods and avoids inaccurate local block deformation models. 3) Realistic methods, i.e., E-Global and T-Global, to lower the computational complexity: These methods can remove a large portion of redundant computations related to the factors $|S(T)|$ (the number of templates in a database) and L (the dimensionality of the feature vector).

Our planned future work includes applying these methods to real DIR applications and other similar applications, and introducing more geometric knowledge to refine the high-dimensional pattern-matching algorithms.

Acknowledgments

The authors would like to thank the anonymous reviewers for their very helpful comments to improve this paper. Zheru Chi is supported by an ASD research grant from The Hong Kong Polytechnic University (Ref. No.: A408).

References

[1] D.A. Adjeroh and M.C. Lee, "On Ratio-Based Color Indexing," *IEEE Trans. Image Processing*, vol. 10, no. 1, pp. 36-48, 2001.
[2] B. Braunmuller, M. Ester, H.-P. Kriegel, and J. Sander, "Multiple Similarity Queries: A Basic DBMS Operation for Mining in Metric Databases," *IEEE Trans. Knowledge and Data Eng.*, vol. 13, no. 1, pp. 79-95, Jan./Feb. 2001.
[3] F. Cesarini, M. Gori, S. Marinai, and G. Soda, "INFORMys: A Flexible Invoice-Like Form-Reader System," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 7, pp. 710-745, July 1998.
[4] J. Chen, C.A. Bouman, and J.C. Dalton, "Hierarchical Browsing and Search of Large Image Databases," *IEEE Trans. Image Processing*, vol. 9, no. 3, pp. 442-455, 2000.
[5] D. Doermann, H. Li, and O. Kia, "The Detection of Duplicates in Document Image Databases," *Proc. Fourth Int'l Conf. Document Analysis and Recognition*, pp. 314-318, 1997.
[6] K. Fan and M. Chang, "Form Document Identification Using Line Structure Based Features," *Proc. Fourth Int'l Conf. Pattern Recognition*, vol. 2, pp. 1098-1100, 1998.
[7] K. Fukunaga and P.M. Narendra, "A Branch and Bound Algorithm for Computing k-Nearest Neighbors," *IEEE Trans. Computers*, vol. 24, no. 7, pp. 750-753, July 1975.
[8] J. Hafner, H.S. Sawhney, W. Equitz, M. Flickner, and W. Niblack, "Efficient Color Histogram Indexing for Quadratic Form Distance Functions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 7, pp. 729-736, July 1995.
[9] J. Hu, R. Kashi, and G. Wilfong, "Document Image Layout Comparison and Classification," *Proc. Sixth Int'l Conf. Document Analysis and Recognition*, pp. 285-288, 1999.
[10] J.J. Hull, "Document Image Similarity and Equivalence Detection," *Int'l J. Document Analysis and Recognition*, vol. 1, no. 1, pp. 37-42, 1998.
[11] D.P. Lopresti, "String Techniques for Detecting Duplicates in Document Databases," *Int'l J. Document Analysis and Recognition*, vol. 2, no. 4, pp. 186-199, 2000.
[12] H. Peng and Q. Gan, "SOCR 1.03: A Handwritten Data Form Producing and Reading System," *Proc. 2000 Int'l Workshop Multimedia Data Storage, Retrieval, Integration, and Applications*, pp. 197-202, 2000.
[13] H. Peng, Z. Chi, W. Siu, and D. Feng, "PageX: An Integrated Document Processing Software for Digital Libraries," *Proc. 2000 Int'l Workshop Multimedia Data Storage, Retrieval, Integration, and Applications*, pp. 203-207, 2000.
[14] H. Peng, F. Long, Z. Chi, D. Feng, and W. Siu, "Document Image Matching Based on Component Blocks," *Proc. Int'l Conf. Image Processing*, pp. 601-604, Sept. 2000.
[15] H. Peng, F. Long, Z. Chi, and W. Siu, "Document Template Matching Based on Component Block List," *Pattern Recognition Letters*, vol. 22, no. 9, pp. 1033-1042, 2001.
[16] J. Puzicha, J.M. Buhmann, Y. Rubner, and C. Tomasi, "Empirical Evaluation of Dissimilarity Measures for Color and Texture," *Proc. Seventh IEEE Int'l Conf. Computer Vision*, vol. 2, pp. 1165-1172, 1999.
[17] R. Safari, N. Narasimhamurthi, M. Shridhar, and M. Ahmadi, "Document Registration Using Projective Geometry," *IEEE Trans. Image Processing*, vol. 6, no. 9, pp. 1337-1341, 1997.
[18] S. Shimotsuji and M. Asano, "Form Identification Based on Cell Structure," *Proc. 13th Int'l Conf. Pattern Recognition*, vol. 3, pp. 793-797, 1996.
[19] L. Tseng and R. Chen, "The Recognition of Form Documents Based on Three Types of Line Segments," *Proc. Fourth Int'l Conf. Document Analysis and Recognition*, vol. 1, pp. 71-75, 1997.
[20] T. Watanabe, Q. Luo, and N. Sugie, "Layout Recognition of Multi-Kinds of Table Form Documents," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 4, pp. 432-445, Apr. 1995.