



A probabilistic method for keyword retrieval in handwritten document images

Huaigu Cao*, Anurag Bhardwaj, Venu Govindaraju

Center for Unified Biometrics and Sensors (CUBS), Department of Computer Science and Engineering, University at Buffalo, Amherst, NY 14260, USA

ARTICLE INFO

Article history:

Received 8 August 2008

Received in revised form 17 January 2009

Accepted 5 February 2009

Keywords:

Word spotting

Information retrieval

Handwriting recognition

ABSTRACT

Keyword retrieval in handwritten document images is a challenging task because handwriting recognition does not perform adequately to produce the transcriptions, specially when using large lexicons. Existing methods build indices using OCR distances or image features for the purpose of retrieval. These alternative methods are complimentary to the traditional approaches that build indices on OCR'ed text. In this paper, we describe an improvement to the existing keyword retrieval (word spotting) methods by modeling imperfect word segmentation as probabilities and integrating these probabilities into the word spotting algorithm. The scores returned by the word recognizer are also converted into probabilities and integrated into the probabilistic word spotting model.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

Keyword retrieval in handwritten document images is a high-level application that relies on document analysis and recognition techniques. There are two common approaches to keyword retrieval from handwritten documents. In the first approach [1–8], image-to-image matching is used. During retrieval, each keyword is converted into a word image. This is done by annotating a small set of word images or collecting the user's handwriting on-line. When a user provides a query word, the similarity between the query and any word image in the database is computed. All of the word images are returned in the decreasing order of the similarities between them and the query. The similarity between two word images is measured as a distance between the two feature vectors computed from the word images. In [1,3], the similarity between the feature vectors of two word images is computed by dynamic time warping (DTW) matching of profile features using various definitions of matching distances [1,9,10,3,11] in the feature space. The GSC-matching method [2,12] is based on bitwise matching of the corresponding GSC features of two word images. Thus, word spotting is a useful alternative when a full fledged handwriting recognition system is not available.

However, word spotting requires on-line matching which is time-consuming. Trade-off between accuracy and speed has to be made in order to scale to large databases. Thus, in order to be

fast matching-based indexing approaches are limited in feature selection and the complexity of matching and training methods. This also limits their scope to applications dealing with a single writer or small lexicons.

In contrast, OCR score-based indexing approaches [13–15] do not face the speed problem. In these methods, the indices are built from OCR scores such as posterior probabilities or feature vector observational likelihoods (probability density) obtained from distances returned by word recognizer. These methods [13–15] perform handwriting recognition followed by an indexing step to keep track of the transcription and other useful information (positions and recognition scores of word images). The similarity between the keyword and another word image is computed using the recognition scores, which are usually the likelihood of the feature space, probabilities, or some other distance-based measurements. One question is whether to adopt a word lexicon. The index for fast retrieval can be built on the results of word level recognition in lexicon-driven mode [14,15]. In this mode, any word that is not in the lexicon cannot be retrieved. Ref. [13] performs recognition at the character level and searches for words in a series of character recognition scores. However, this approach is once again difficult and time-consuming which does not scale to larger data sets. We have taken a word-lexicon-driven method and get affected by the out-of-vocabulary (OOV) problem.

We have improved the OCR score-based indexing method by integrating word segmentation probabilities into the retrieval similarity metric. Word spotting methods this far have assumed perfect word segmentation: word images are given by word segmentation algorithm, and the ranks of word images are obtained by sorting the word recognition scores. However it is unrealistic to expect perfect word segmentation in unconstrained handwriting given the variation in the gap sizes between words. The performance of

* Corresponding author.

E-mail addresses: hcao@bun.com (H. Cao), ab94@buffalo.edu (A. Bhardwaj), govind@buffalo.edu (V. Govindaraju).

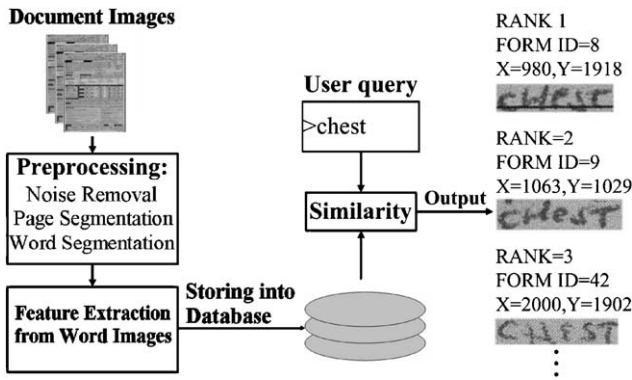


Fig. 1. Diagram of the keyword spotting system.

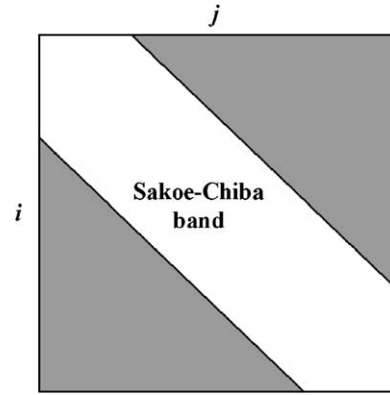


Fig. 2. Sakoe-Chiba band.

word spotting can be improved by modeling the word segmentation probabilities. In this paper, we describe a probabilistic model of word spotting that integrates word segmentation probabilities and word recognition probabilities. The word segmentation probabilities are obtained by modeling the conditional distribution of multivariate distance features of word gaps. The word recognition results are also represented by a probabilistic model. The modeling of the word recognition probabilities is obtained from the distances returned by the word recognizer (Fig. 1).

2. Background in handwritten keyword retrieval

2.1. Image-to-image matching—word spotting

Word spotting was initially proposed as an alternative approach for indexing and retrieving handwritten documents, that is one could search handwritten document images without using a handwriting recognizer. In order to search for a keyword, the user needs to write a copy of the keyword (a word template) and provide the word image as the query. One could also obtain the word templates by labeling a training set. The system executes the query by computing the distance between the query template and each word image in the document images.

DTW-based keyword spotting: In the DTW-based method [1,3,11], the following preprocessing steps are commonly used.

1. Word segmentation is performed and the background of every word image is cleaned by removing irrelevant connected components from other words that reach into the word's bounding box.
2. Inter-word variations such as skew and slant angle are detected and eliminated.
3. The bounding box of any word image is cropped so that it tightly encloses the word.
4. The baseline of word images is normalized to a fixed position by padding extra rows to the images.

A normalized word image is represented by a multivariate time series composed of features from each column of the word image. These features include projection profile, upper/lower word profile, and number of background-to-foreground transitions.

1. Projection profile. The projection profile of a word image is composed of the sum of foreground pixels in each column.
2. Upper/lower profiles. The upper profile of a word image is made of the distances from the upper boundary to the nearest foreground pixels in each column.
3. Background-to-foreground transitions. The number of background pixels whose right neighboring pixels are foreground

pixels is taken as the number of background-to-foreground transitions of the column.

Suppose two word images w_A and w_B are represented by $\{f_A(1), f_A(2), \dots, f_A(l_A)\}$ and $\{f_B(1), f_B(2), \dots, f_B(l_B)\}$, respectively, where $f_A(i)$ is the feature vector of the i -th column of image w_A , $f_B(j)$ is the feature vector of the j -th column of image w_B , and l_A and l_B are the lengths of w_A , w_B , respectively. Then the DTW matching distance of w_A and w_B is given by the recurrence equation

$$DTW(i,j) = \min \left\{ \begin{array}{l} DTW(i-1,j) \\ DTW(i-1,j-1) \\ DTW(i,j-1) \end{array} \right\} + d(i,j) \quad (1)$$

where $d(i,j)$ is the square of the Euclidean distance between $f_A(i)$ and $f_B(j)$.

The time complexity of the DTW algorithm is in $O(l_A \cdot l_B)$. In order to reduce the computation and prevent pathological warping, a global path constraint like the Sakoe-Chiba band can be applied to force the paths to stay close to the diagonal of the DTW matrix. In Fig. 2, the dynamic programming range of (i,j) is restricted within a band along the diagonal of the (i,j) matrix which is called the Sakoe-Chiba band.

The matching error of $f_A(i)$ and $f_B(j)$ is given by $(1/l)DTW(l_A, l_B)$ where l is the length of the warping path recovered by DTW. The word images are ranked in the increasing order of the matching errors to the template image.

The DTW-based method has been tested on George Washington's manuscripts (CIIR, University of Massachusetts [1,11]). The performance of keyword spotting was evaluated using the mean average precision (MAP) measure [16]:

1. For each query, check the returned word images starting from rank 1. Whenever a relevant word image is found, keep track of the precision of the word images from the one with rank 1 to the current one. The average value of the recorded precisions for the query is taken as the average precision (AP) of the query.
2. The mean value of the AP of all of the queries is the MAP of the test.

A MAP of 40.98% on 2372 word images of good quality and a MAP of 16.50% on 3262 word images of poor quality was reported on George Washington's manuscripts [3].

GSC feature-based keyword spotting: In the GSC feature-based method [2,12], a word image is represented by 1024 bits of the GSC features corresponding to the gradient (192 bits), structural (192 bits) and concavity (128 bits) features. A word image is divided into 32 regions (8×4) and 16 binary GSC features are extracted from each region. The gradient features are obtained by thresholding the results of Sobel edge detection in the 12 directions. The structural

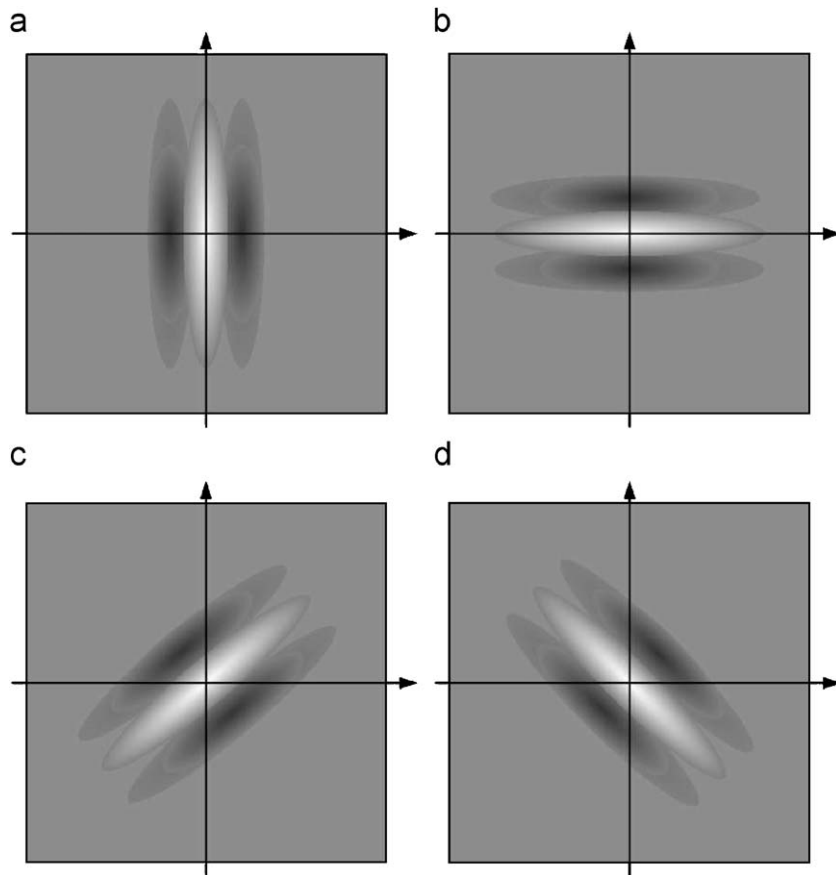


Fig. 3. The Gabor filters of four different orientations. (a) Vertical Gabor filter; (b) horizontal Gabor filter; (c) diagonal Gabor filter; (d) anti-diagonal Gabor filter.

features consist of the presence of corners, diagonal lines and vertical and horizontal lines in the gradient image, as determined by the 12 rules. The concavity features include the direction of bays, presence of holes, and large vertical and horizontal strokes.

The similarity of two word images is measured by the bitwise matching of the respective GSC feature vectors of the two images. The dissimilarity of two GSC feature vectors X and Y is defined as

$$D(X, Y) = \frac{1}{2} - \frac{S_{11}S_{00} - S_{10}S_{01}}{2\sqrt{(S_{10} + S_{11})(S_{01} + S_{00})(S_{11} + S_{01})(S_{00} + S_{10})}} \quad (2)$$

where S_{00} , S_{01} , S_{10} , and S_{11} are the numbers of 0-to-0, 0-to-1, 1-to-0, and 1-to-1 matchings from X to Y , respectively. For example, the numbers of 0-to-0, 0-to-1, 1-to-0, and 1-to-1 matchings between “0110110” and “0101001” are 1, 2, 3, and 1, respectively.

The GSC method was tested on 9312 word images of four words (“been”, “Cohen”, “Medical”, and “referred”) written by 776 individuals. Each word was written three times by each individual. One of the three word images for every word written by any person is taken as a query template, and the remaining are taken for test. The performance of keyword spotting is evaluated by the recall and precision at different number of top matchings. When the number of top matchings of a query equals the number of relevant images, the recall value equals the precision value and is referred to as R-precision.

The results of both the GSC-based method and the DTW-based method are reported in [2]. The R-precision values of the above four queries using the GSC-based method are 45.45%, 56.59%, 54.11%, and 62.04%, respectively. The R-precision values of the above four queries using the DTW-based method are 35.53%, 38.65%, 44.39%, and 55.23%, respectively. Although the above results are obtained from a data set of multiple writers, the size of the word lexicon is

very small (containing only four words) and therefore the data set is not truly unconstrained.

2.2. Keyword retrieval using word recognizers

Word spotting methods are useful when one does not have a handwriting recognizer. On the other hand, the word matching, which is essential to word spotting, can be thought of as a prototype of word recognizer, although its performance is considerably poorer than that of a well trained word recognizer. But handwriting recognition remains very challenging task due to the wide variations in the handwriting. Thus matching against a single template is not a robust approach.

The advantage of word recognizer-based word retrieval over simple word matching was observed in our prior work [13] by comparing the performance of DTW-based word spotting method with the recognition-based keyword retrieval method. In [13], we used the word recognition probability to define the similarity between query word w composed of n characters: $w = c_1 c_2 \dots c_n$ and the word image I_w . Suppose a segmentation of I_w is represented by

$$s_{I_w} = [V_1 V_2 \dots V_n] \quad (3)$$

where V_i ($1 \leq i \leq n$) is the feature vector of the i -th character image, then the similarity between w and I_w is given as

$$\begin{aligned} \text{sim}(w, I_w) &= \underset{[V_1 V_2 \dots V_n]}{\text{argmax}} [\Pr(c_1, \dots, c_n | V_1, \dots, V_n)]^{1/n} \\ &\approx \underset{[V_1 V_2 \dots V_n]}{\text{argmax}} \left[\prod_{i=1}^n \Pr(c_i | V_i) \right]^{1/n} \end{aligned} \quad (4)$$

Table 1

Six query words used in the keyword retrieval tests in [13].

Pain	Supine	Physical
Chest	Back	Pulse

where $\Pr(c_1, \dots, c_n | V_1, \dots, V_n)$ is the posterior probability that I_w is w , i.e., the probability that I_w is w , conditioned by observation $[V_1 V_2 \dots V_n]$.

The character images are bilinearly interpolated to the same size. Histogram features [17] are computed from the output of Gabor wavelet transform on each character image. We apply the Gabor wavelet at two scales. At each scale, we perform the Gabor filtering in four orientations: horizontal, vertical and two diagonals. A Gabor filter is a band-pass filter with specified central frequency and central angle and uses Gaussian window. The spatial domain responses of the Gabor filters of the above four orientations are illustrated in Fig. 3. We divide the filtered image at the low-resolution scale into 4 bins, and divide the filtered image at the high-resolution scale into 16 bins. We compute the sums of positive and negative real parts, respectively, from each bin. Thus we obtain 160 features per character image. The support vector machine (SVM) with the RBF kernel is used for character recognition. The posterior probability $\Pr(C_i | V_i)$ in Eq. (4) is modeled as a function of the decision value of the SVM classifier and estimated from the training set [18].

We searched a small data set of 12 documents including 101 words written in English for the six keywords listed in Table 1. The MAP obtained by our method is 67.1%. The MAP obtained on the same data set and queries using DTW matching is only 12.6%.

3. Keyword retrieval, an important component of the search engine for off-line handwriting

3.1. A search engine for off-line handwriting

A handwritten document retrieval system is presented in our prior work [19]. The goal of document retrieval is to search for “documents” that are relevant to the user query, as opposed to keyword retrieval that aims at searching for keywords. In document retrieval, we use standard indexing techniques such as TF-IDF to build indices from the documents. The major challenge in retrieving handwritten documents is the difficulty of computing the term frequency (TF) due to recognition errors. Our approach is to maintain an N -best list of the handwriting segmentation and recognition hypotheses, and estimate the TF using each result of the N -best list. The final TF is defined as a weighted sum of all the above TFs where the weights are the probabilities of validity of the segmentation and recognition hypothesis.

In the classic vector model [20], the documents are represented a vector space. Given the word vocabulary $\{t_i\}$, $1 \leq i \leq N$, the TF of document d_j is defined by

$$tf_{ij} = \frac{freq_{ij}}{L}, \quad i = 1, \dots, N \quad (5)$$

where $freq_{ij}$ is the number of occurrences of term t_i in document d_j and L is the total number of occurrences of all terms in document d_j , i.e., the length of d_j . For example, in a document d_j of 1000 words, term $t_i = \text{“diseases”}$ occurs three times, then raw TF $freq_{ij} = 3$ and TF $tf_{ij} = 0.003$. Thus document d_j is represented by vector $[tf_{1j}, tf_{2j}, \dots, tf_{Nj}]$. The raw frequency $freq_{ij}$ is estimated using

$$E(freq_{ij}) = \sum_{\vec{w}} \Pr(\vec{w} | \vec{\sigma}) \cdot \sum_{\vec{\tau}} \Pr(\vec{\tau} | \vec{w}) \cdot \#_{t_i}(\vec{\tau}) \quad (6)$$

where $\vec{\sigma}$ is the observation sequence of the image features of the text in document d_j , \vec{w} is a word-level segmentation of sequence $\vec{\sigma}$,

$\vec{\tau}$ is a sequence of terms, $\Pr(\vec{w} | \vec{\sigma})$ is the probability that \vec{w} is a valid segmentation, $\Pr(\vec{\tau} | \vec{w})$ is the word sequence recognition probability, and $\#_{t_i}(\vec{\tau})$ is the number of term t_i occurring in sequence $\vec{\tau}$. Given word segmentation probabilities, word recognition likelihood and the language model (n -gram), Eq. (6) is solved using dynamic programming.

A simple way of estimating $E(freq_{ij})$ by using the transcription produced by handwriting recognition under-estimates the TFs, and thus leads to poor performance. Eq. (6) utilizes the N -best information and can improve both recall and precision of the system. For more details about the above-mentioned IR system refer to [19].

When we search for documents relevant to our query, usually we also want to get the positions of the query words and highlight them in documents, because we may only want to read upon the context around the query words. Text retrieval systems usually keep track of the positions of all the term in the indexing file. In our application, since the word segmentation is not perfect, we can only obtain hypotheses of word images. In addition to the positions, we also need to keep track of the similarities between word images and terms. The similarities can be defined and computed with very little effort given that the indexing of document retrieval has been done.

3.2. Word spotting using segmentation probabilities

3.2.1. Word spotting model

Given a series of consecutive connected components c_1, c_2, \dots, c_n and word image w represented by c_i, c_{i+1}, \dots, c_j ($1 \leq i, j \leq n$), the similarity between w and a query word q is defined by

$$\text{sim}(w, q) = \sigma_{i-1} \cdot (1 - \sigma_i) \cdot \dots \cdot (1 - \sigma_{j-1}) \cdot \sigma_j \cdot \Pr(q|w) \quad (7)$$

where σ_k ($1 \leq k \leq n-1$) is the probability that the gap between c_{k-1} and c_k is a valid word gap, and $\Pr(q|w)$ is the probability that word image w is an image of word q . For the convenience of computation, we assume that the preceding blank of the left-most connected component and the succeeding blank of the right-most connected component are always valid word gaps (σ_0 and σ_n). The probability of validity of these two gaps is always 1, i.e., $\sigma_0 = \sigma_n = 1$. Here we assume the gaps are independent and thus the word segmentation probability is $\sigma_{i-1} \cdot (1 - \sigma_i) \cdot \dots \cdot (1 - \sigma_{j-1}) \cdot \sigma_j$.

The size of the space to store the index (word location and similarities) can be reduced by applying the constraints on the number of connected components within a word image and minimum similarity:

Given a series of consecutive connected components c_1, c_2, \dots, c_n ,

For i from 1 to n

For j from i to $\min(i + C_{max} - 1, n)$

If the similarity $\text{sim}(c_i \dots c_j, q) > \text{sim}_{min}$

Then store the document number, coordinates of the word image and the similarity into index.

C_{max} is the maximum number of connected components within a word image. sim_{min} is the minimum similarity that can be stored into the index. We assume $C_{max} = 16$ and $\text{sim}_{min} = 0.03\%$ in our experiment.

3.2.2. Estimating word segmentation probability

Word segmentation is defined as the process of segmenting a line into words. In handwritten lines, the space between words is uneven. Moreover, the space of the same size may be present between words, and between characters within a word. Such cases arise due to differences in writing styles, and the limited blank space left for writing.

In our word segmentation method, the word segmentation probabilities are estimated from distance-based features. The gap between

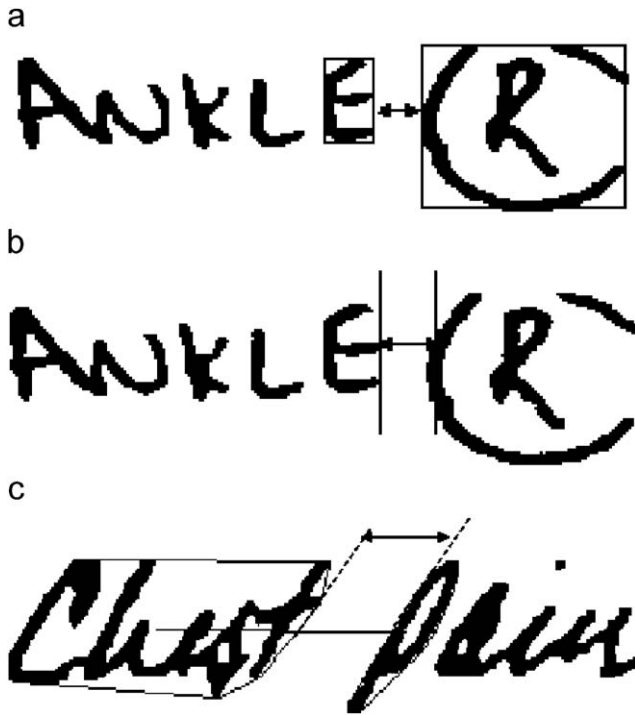


Fig. 4. Three features representing a gap between two consecutive connected components. (a) Euclidean distance; (b) run length distance; (c) convex hull distance.

any two consecutive connected components is represented by three distance features:

1. *Euclidean distance*: This feature is defined as the horizontal distance between the bounding boxes of the two consecutive connected components of the line image (Fig. 4(a)).
2. *Minimum run length*: This feature represents the minimum horizontal white run length distance between the two adjacent connected components of the line image.
3. *Convex hull distance*: We compute the convex hulls of two consecutive connected components and draw a line connecting the mass centers of the two convex hulls. The Euclidean distance between points at which this line crosses the two convex hulls is defined as the convex hull distance of the two adjacent components.

To eliminate the effect by the variation in the text sizes, we normalize the extracted features by dividing them by the average height of all components in the same line.

The segmentation probability of a gap g is given by the Bayes' rule

$$\sigma_g = \frac{\Pr(g|f_{1,g}, f_{2,g}, f_{3,g})}{\Pr(g)p(f_{1,g}, f_{2,g}, f_{3,g}|g) + \Pr(\bar{g})p(f_{1,g}, f_{2,g}, f_{3,g}|\bar{g})} \quad (8)$$

where $\Pr(g)$ and $\Pr(\bar{g})$ are the prior probabilities of valid gaps and non-valid gaps, respectively. $f_{1,g}$, $f_{2,g}$ and $f_{3,g}$ are three features of g .

Given a set of gap features with the annotation of “valid” and “non-valid”, we estimate $\Pr(g)$, $\Pr(\bar{g})$, $p(f_{1,g}, f_{2,g}, f_{3,g}|g)$ and $p(f_{1,g}, f_{2,g}, f_{3,g}|\bar{g})$ as follows. $\Pr(g)$ and $\Pr(\bar{g})$ are estimated from the ratio of the numbers of valid and non-valid gaps in the training set:

$$\Pr(g) = \frac{\#\{\text{valid gaps}\}}{\#\{\text{valid gaps}\} + \#\{\text{non-valid gaps}\}} \quad (9)$$

$$\Pr(\bar{g}) = 1 - \Pr(g) \quad (10)$$

$p(f_{1,g}, f_{2,g}, f_{3,g}|g)$ and $p(f_{1,g}, f_{2,g}, f_{3,g}|\bar{g})$ are estimated non-parametrically using Parzen window technique with a Gaussian kernel function.

3.2.3. Estimating word recognition probability

In our system, the matching distance between a word image and a word is obtained by the word recognition algorithm of [21]. In this word recognition method, for any word image, all possible locations of the ligatures connecting two characters are identified by heuristic analysis of the concavity and convexity of the contour image. Then the word image can be divided into several pieces. By assuming that a character consists of at most four consecutive pieces, we can create a series of hypotheses of character images. Various features including the directions along the image contour are computed from each hypothesis of character image. The distance between the hypothesis H and any character class C , denoted by $D(H, C)$, is defined by the Euclidean distance between the feature vector of the hypothesis and the centroid of all the training samples in that character class. The word-level matching distance is defined using the character-level Euclidean distances. The distance between any word image w and any word with n characters: $C_1 C_2 \dots C_n$ is computed by enumerating all possible segmentations of w such that

$$s(w, C_1 C_2 \dots C_n) = \min_{H_1 H_2, \dots, H_n} \sum_{i=1}^n D^2(H_i, C_i) \quad (11)$$

where H_1, H_2, \dots, H_n are one of the possible segmentations of w into n hypotheses of character images.

The square distance $s(w, t_i)$ between any word image w and a term t_i in the lexicon is converted into word recognition probabilities using the universal background model (UBM) [22]. In the background model, the posterior probability of the word recognition is computed by Bayes' rule:

$$\Pr(w = t_i | s(w, t_i)) = \frac{\Pr(w = t_i) p_{t_i}(s(w, t_i) | w = t_i)}{\Pr(w = t_i) p_{t_i}(s(w, t_i) | w = t_i) + \Pr(w \neq t_i) p_{t_i}(s(w, t_i) | w \neq t_i)} \quad (12)$$

where $p_{t_i}(s(w, t_i) | w = t_i)$ is the likelihood of the true matching score when the word is t_i , $p_{t_i}(s(w, t_i) | w \neq t_i)$ is the likelihood of the false matching score when the word is t_i , and $\Pr(w = t_i)$, $\Pr(w \neq t_i)$ are the prior probabilities of true and false matching of t_i , respectively.

We need a term specific training set for every term to learn the background model. This is a drawback in applications using large number of terms. The UBM is an alternative approach that solves this problem. In the UBM, we use a single background model for all of the terms. The true matching probability is given by

$$\Pr(\text{True_Matching} | s) = \frac{\Pr(\text{True_Matching}) p(s | \text{True_Matching})}{\Pr(\text{True_Matching}) p(s | \text{True_Matching}) + \Pr(\text{False_Matching}) p(s | \text{False_Matching})} \quad (13)$$

where s is a matching score, and $\Pr(\text{True_Matching})$ and $\Pr(\text{False_Matching})$ are the prior probabilities of true matching and false matching, respectively, and $p(s | \text{True_Matching})$, $p(s | \text{False_Matching})$ are the likelihoods of the score of true

$p(f_{1,g}, f_{2,g}, f_{3,g}|g)$ is the probability density of the features of valid gaps. $p(f_{1,g}, f_{2,g}, f_{3,g}|\bar{g})$ is the probability density of the features of non-valid gaps.

matching and false matching, respectively. $\Pr(\text{True_Matching})$, $\Pr(\text{False_Matching})$, $p(s|\text{True_Matching})$, and $p(s|\text{False_Matching})$ are estimated from the scores of all of the terms.

We model $p(s|\text{True_Matching})$ and $p(s|\text{False_Matching})$ as gamma distributions. Actually, the matching score s is a squared sum of distances between character-level feature vectors and the centers of clusters in the training features. In other words,

$$s = \sum_{l=1}^L D_l^2 \tag{14}$$

where D_l is a character matching distance. If we assume all the clusters of the training feature vector space are independent normal distributions, then the squared sum of the distances can be modeled as a gamma distribution. The probability density function of the gamma distribution can be represented by

$$f_s(s; k, \theta) = s^{k-1} \frac{e^{-s/\theta}}{\theta^k \Gamma(k)}, \quad s > 0 \text{ and } k, \theta > 0 \tag{15}$$

where $\Gamma(k)$ is the gamma function:

$$\Gamma(k) = \int_0^\infty x^{k-1} e^{-x} dx \tag{16}$$

If k is a positive integer, then $\Gamma(k) = (k - 1)!$. There is no closed-form solution for the maximum likelihood estimation of k and θ [23]. However, we can use a simple way to estimate the gamma distribution. First we can prove that the mean and variance of the gamma distribution are $k \cdot \theta$ and $k \cdot \theta^2$, respectively. Then, given N true matching scores s_1, s_2, \dots, s_N , we can compute the ML estimation of mean and variance:

$$\begin{cases} \bar{\mu} = \frac{1}{N} \sum_{i=1}^N s_i \\ \bar{\sigma}^2 = \frac{1}{N} \sum_{i=1}^N (s_i - \bar{\mu})^2 \end{cases} \tag{17}$$

Let $\bar{k} \cdot \bar{\theta} = \bar{\mu}$ and $\bar{k} \cdot \bar{\theta}^2 = \bar{\sigma}^2$, then

$$\begin{cases} \bar{k} = \frac{\bar{\mu}^2}{\bar{\sigma}^2} \\ \bar{\theta} = \frac{\bar{\sigma}^2}{\bar{\mu}} \end{cases} \tag{18}$$

4. Experimental results

4.1. Data collection

Our corpus consists of 1125 scanned images of the New York State Pre-hospital Care Report (PCR) forms. We index the handwritten text in these form images for keyword retrieval. Our task is very challenging because the images are scanned from the carbon copies which are the only available existing copies of the forms. The degradation of the text in the PCR forms can be seen from Fig. 5. We obtain an English word vocabulary and a bi-gram language model from the ground-truth word of 783 PCR forms. We use the vocabulary in our recognition-based keyword retrieval method. The bi-gram is only used in other applications like document retrieval. The number of words in the vocabulary is 4570. We use the rest of the corpus including 342 PCR forms to test our keyword retrieval method. In our tests, we search the 342 PCR forms for the queries listed

in Table 2. The handwritten word recognition system [21] is trained using 21 054 English character images extracted from the handwritten image data collected by the USPS.

4.2. Preprocessing

First we detect and remove the skew of every PCR form image as follows.

1. We manually de-skew a form and take it as a template. Two regions of pre-printed headlines are cropped from the template as anchors.
2. The positions of two anchoring regions in any test image are found by cross-correlation.
3. The skew angle of the test image is obtained by the relative skewing between the test image and the template. We de-skew the image by rotating to the opposite direction.

By aligning the test image to the template image, we can also obtain the position of each form cell containing a line of text. The de-skewing and page segmentation method using template-matching works well on the PCR form images since they have a fixed layout and are scanned at the same resolution. Our approach is applicable to other types of forms as well.

We use the Markov random fields (MRF)-based document image preprocessing algorithm [24] to binarize the form image and remove the grid lines from the image. Assuming the binarized objective image is x and the grayscale image is y , we solve the maximum a posteriori (MAP) estimation $\hat{x} = \text{argmax}_x \Pr(x|y)$ using the MRF. An example of binarization and line removal result is shown in Fig. 6. The MRF-based preprocessing method improves the word recognition accuracy from 18.7% (obtained by the PCR form preprocessing algorithm in [25]) to 28.6%.

4.3. Evaluation metrics

The performance of word spotting is evaluated using the precisions at 11 recall levels (0, 0.1, ..., 1). We also use single value measures such as the MAP [16] to evaluate the word spotting performance. The MAP is computed as follows:

1. Suppose Q is a set of queries. For each query $q \in Q$, check the returned word images starting from rank 1 down to the lowest ranks. Whenever a relevant word image is found, keep track of the precision of the word images from the one with rank 1 to the current one. The AP of a given query q is weighted sum of the recorded precisions:

$$AP(q) = \frac{\sum_{1 \leq r \leq N_q, Rel(r) \text{ is true}} Prec(r)}{R_q} \tag{19}$$

where N_q is the number of word images returned, R_q is the number of relevant documents, $Prec(r)$ is the precision of top- r returned word images, and $Rel(r)$ is a Boolean function indicating if the word image returned at rank r is relevant.

2. The MAP of all the queries $q \in Q$ is

$$MAP = \frac{\sum_q AP(q)}{\text{Number of queries in } Q} \tag{20}$$

4.4. Keyword retrieval experiments

We compare four different keyword retrieval methods on the PCR corpus. These methods include the DTW matching [11], GSC feature matching [12], and two methods based on the handwriting

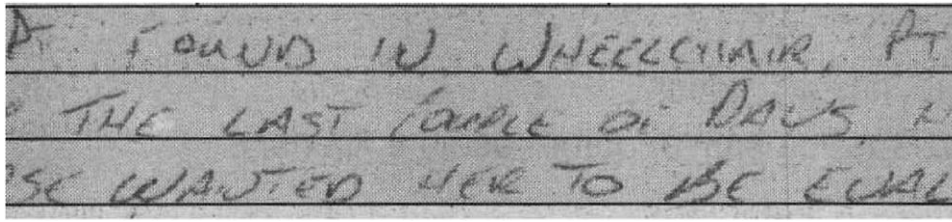


Fig. 5. The text in a PCR form.

Table 2
The query words used in the keyword retrieval tests.

Ambulate	Arthritis	Ankle	Back	Blind	Blood
Breath	Cancer	Cardiac	Chest	Dementia	Dizzy
Dizziness	Diabetes	Emesis	Foot	Fracture	Glucose
Head	Hurts	Lung	Monitor	MRI	Neck
Pain	Rib	Shoulder	Syncope	Tender	Trachea
Trauma	Vitals	Wrist			

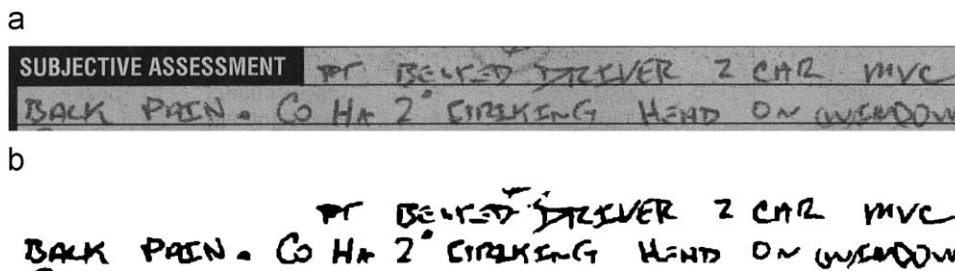


Fig. 6. An example of the binarization and line removal result. (a) The original grayscale image. (b) The binarized image. Grid lines are removed and broken strokes are fixed.

recognition distance [21]: one using the simple segmentation and the other using more complicated probabilistic segmentation (the proposed method).

- Test 1 DTW matching: In the DTW matching method, word segmentation is performed using the gap labeling probabilities described in Section 3.2. First the connected components are extracted from the binarized image. Then these connected components are grouped into word images using a cutoff threshold C_{gap} of the maximum word gap probability. For any two adjacent connected components c_k and c_{k+1} within a word image, the word gap probability $\sigma_k < C_{gap}$. The cutoff threshold $C_{gap} = 0.297$ in our test. The recall rate of the gap classification reached maximum (0.394) when $C_{gap} = 0.297$. Each word image is represented by a time sequence of multivariate features. The upper and lower profiles of the word are computed. We randomly select three word images of each query from our training set of 783 PCR forms to serve as the matching templates. The recursive definition of the matching distance

$$S_{w,t_i}^{WR} = \frac{\Pr(\text{True_Matching}|s)}{\Pr(\text{True_Matching})p(s|\text{True_Matching}) + \Pr(\text{False_Matching})p(s|\text{False_Matching})} \quad (23)$$

in Eq. (1), i.e.,

$$D_{w,T}^{DTW}(i,j) = \min \left\{ \begin{array}{l} D_{w,T}^{DTW}(i-1,j) \\ D_{w,T}^{DTW}(i-1,j-1) \\ D_{w,T}^{DTW}(i,j-1) \end{array} \right\} + d(i,j) \quad (21)$$

is used to compute the similarity between any word image w in the test set and template image T . For each query, we choose

the minimum of the three distances as the best matching distance. The word images are ranked in the increasing order of the matching distance.

- Test 2 GSC matching: In the GSC-matching test, we use the word segmentation results and the matching templates we obtain from Test 1. Every word image is represented using the 1024-bit GSC features [12]. The dissimilarity between word image w and template T are computed using Eq. (2), i.e.,

$$D_{w,T}^{GSC} = \frac{1}{2} - \frac{S_{11}S_{00} - S_{10}S_{01}}{2\sqrt{(S_{10} + S_{11})(S_{01} + S_{00})(S_{11} + S_{01})(S_{00} + S_{10})}} \quad (22)$$

where S_{00} , S_{01} , S_{10} , and S_{11} are the numbers of 0-to-0, 0-to-1, 1-to-0, and 1-to-1 matches from w to T , respectively. We also choose the minimum of the three dissimilarities for each query as the best matching dissimilarity.

- Test 3 word recognition distance matching (simple segmentation): In this test, we also use the word segmentation results we obtain from Test 1. The similarity that we use in word matching is the word recognition probability converted from the word recognition distance using Eq. (13), i.e.,

where s is the square matching distance between word image w and term t_i .

- Test 4 word recognition distance matching (probabilistic segmentation): In this test, word segmentation is also performed using the gap labeling probabilities described in Section 3.2. We take all the hypotheses of word images $w = c_i, c_{i+1}, \dots, c_j$ such that

$$\sigma_{i-1} \cdot (1 - \sigma_i) \cdot \dots \cdot (1 - \sigma_{j-1}) \cdot \sigma_j > 0.001 \quad (24)$$

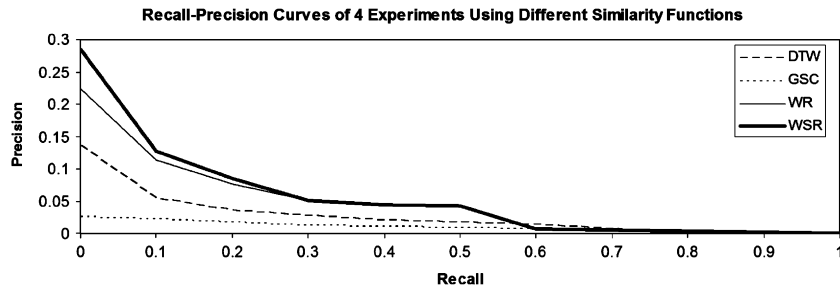


Fig. 7. Precision-recall curves of the four keyword retrieval tests on 342 PCR forms.

Table 3

The mean average precisions (MAP) of the four keyword retrieval tests on 342 PCR forms.

Similarity	D^{DTW}	D^{GSC}	S^{WR}	S^{WSR}
MAP	2.2%	0.9%	4.2%	4.7%

Table 4

Ranks obtained by the four keyword retrieval methods.

Similarity	D^{DTW}	D^{GSC}	S^{WR}	S^{WSR}
# of Rank 1 queries	11	5	12	13
# of Rank 2 queries	5	3	13	10
# of Rank 3 queries	10	13	5	7
# of Rank 4 queries	7	12	3	3

where c_1, c_2, \dots, c_n are the connected components in word image w . The word matching similarity is defined by Eq. (24), i.e.,

$$S_{w,t_i}^{WSR} = \sigma_{i-1} \cdot (1 - \sigma_i) \cdot \dots \cdot (1 - \sigma_{j-1}) \cdot \sigma_j \cdot \Pr(\text{True_Matching}|s) \quad (25)$$

4.5. Analysis of keyword retrieval results

Table 3 shows the MAP of the retrieved results of the four experiments. Fig. 7 shows the AP of all the queries when the recall rate = 0, 0.0, ..., 1.

From Table 3 and Fig. 7 we can see both of the handwriting recognition distance matching methods (S^{WR} and S^{WSR}) have higher performance than the image-to-image matching methods (D^{DTW} and D^{GSC}). Due to the use of word segmentation probabilities, S^{WSR} works better than S^{WR} . By comparing the two word-to-word matching methods, we can see that D^{DTW} works better than D^{GSC} . Due to lack of matching templates, the approach [11] using low-dimensionality word profile features and the elastic matching is more reliable than the approach [12] using the high-dimensionality GSC features and rigid matching. We also need to note that the image-to-image matching approaches are not intended for the unconstrained handwriting which is written by large number of authors. Due to the variation in the style of handwriting by different people, a few word templates are not sufficient for matching. But in our experiments, the DTW approach provides a surprisingly high performance for the degraded PCR data set.

We also provide a per-query comparison of the four methods. First we rank the four methods from 1 to 4 using their MAP values for each query. The numbers of rank 1 to rank 4 queries are listed in Table 4. For example, there are 11 queries in which the DTW method ranks 1st among all of the four methods, according to Table 4. The handwriting recognition-based methods using the S^{WR} and S^{WSR} similarities have the most rank 1 and rank 2 queries. Thus they are more reliable than the other two methods.

5. Conclusion

In this paper we present a novel keyword retrieval method for the handwritten document images. Unlike the existing approaches using the image-to-image matching-based approaches, we use the word recognition distances to improve the word matching accuracy. We estimate the probabilities of word boundary segmentation using the distances between connected components and combine the segmentation and recognition distances to create a probabilistic word matching similarity. We show the improvement obtained by our approach by comparing the image-to-image matching approaches [11,12] with ours. The two recognition-based similarity functions S^{WR} and S^{WSR} outperform the image-to-image matching approaches in both the MAP of all the queries and the number of queries in which a method performs better than all others. The use of word segmentation probabilities in the similarity measurement improves the MAP from 4.2% to 4.7%.

Although the recognition-based approach shows the advantage over the image-to-image matching methods, we may notice that our method does not always have the highest MAP in every query. This suggests the future works can be done to improve the overall performance by combining multiple systems using different image features and similarity measurements. System combination may also effectively fix the intrinsic drawbacks of every single system. For example, we can use the recognition-based method to index the common words for higher performance, and use the image-to-image matching method to search for those OOV keywords.

References

- [1] S. Kane, A. Lehman, E. Partridge, Indexing George Washington's handwritten manuscripts, CIIR Technical Report MM-34, Center for Intelligent Information Retrieval, University of Massachusetts Amherst, 2001, pp. 1–20.
- [2] B. Zhang, S.N. Srihari, C. Huang, Word image retrieval using binary features, Document Recognition and Retrieval XI, vol. 5296, SPIE, Greenbelt, MD, 2004, pp. 45–53.
- [3] R. Manmatha, T.M. Rath, Indexing of handwritten historical documents—recent progress, in: Symposium on Document Image Understanding Technology (SDIUT), 2003, pp. 77–85.
- [4] A. Jain, A. Nambodiri, Indexing and retrieval of on-line handwritten documents, in: Proceedings of the International Conference on Document Analysis and Recognition, 2003, pp. 655–659.
- [5] T. Kwok, M. Perrone, G. Russell, Ink retrieval from handwritten documents, in: Proceedings of the Second International Conference on Data Mining, Financial Engineering, and Intelligent Agents, 2000, pp. 461–466.
- [6] S. Uchiashi, L. Wilcox, Automatic index creation for handwritten notes, in: Proceedings of the International Conference on Acoustic, Speech and Signal Processing, 1999, pp. 3453–3456.
- [7] S. Marinai, M. Marino, G. Soda, Indexing and retrieval of words in old documents, in: Proceedings of the International Conference on Document Analysis and Recognition, 2003, pp. 223–227.
- [8] J. Edwards, Y.W. Teh, D.A. Forsyth, R. Bock, M. Maire, G. Vesom, Making Latin manuscripts searchable using HMMs, in: Proceedings of Neural Information Processing Systems, 2004, pp. 385–392.

- [9] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (2002) 509–522.
- [10] J.L. Rothfeder, S. Feng, T.M. Rath, Using corner feature correspondences to rank word images by similarity, in: *Document Image Analysis and Retrieval Workshop (DIAR'03)*, 2003, pp. 30–35.
- [11] T.M. Rath, R. Manmatha, Word spotting for historical documents, *International Journal on Document Analysis and Recognition* 9 (2007) 139–152.
- [12] S.N. Srihari, G.R. Ball, Language independent word spotting in scanned documents, in: *International Conference on Asian Digital Libraries*, 2008, pp. 134–143.
- [13] H. Cao, V. Govindaraju, Template-free word spotting in low-quality manuscripts, in: *the Sixth International Conference on Advances in Pattern Recognition (ICAPR)*, vol. 5296, Calcutta, India, 2007, pp. 45–53.
- [14] T.M. Rath, R. Manmatha, V. Lavrenko, A search engine for historical manuscript images, in: *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2004, pp. 369–376.
- [15] N.R. Howe, T.M. Rath, R. Manmatha, Boosted decision trees for word recognition in handwritten document retrievals, in: *Proceedings of the SIGIR*, 2005, pp. 377–383.
- [16] R.A. Baeza-Yates, B.A. Ribeiro-Neto, *Modern Information Retrieval*, ACM Press, Addison-Wesley, 1999 pp. 79–80.
- [17] X. Wang, X. Ding, C. Liu, Optimized Gabor filter based feature extraction for character recognition, in: *Proceedings of the International Conference on Pattern Recognition*, 2002, pp. 223–226.
- [18] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, software available at (<http://www.csie.ntu.edu.tw/~cjlin/libsvm>), 2001.
- [19] H. Cao, V. Govindaraju, Processing and retrieving handwritten medical forms, in: *Proceedings of the Digital Government Conference (DG.O)*, 2008, pp. 371–372.
- [20] G. Salton, A. Wong, C.S. Yang, A vector space model for automatic indexing, *Communications of the ACM* 18 (11) (1975) 613–620.
- [21] G. Kim, V. Govindaraju, A lexicon driven approach to handwritten word recognition for real-time applications, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19 (1997) 366–379.
- [22] D.A. Reynolds, T.F. Quatieri, R.B. Dunn, Speaker verification using adapted Gaussian mixture models, *Digital Signal Processing* 10 (1–3) (2000) 19–41.
- [23] S.C. Choi, R. Wette, Maximum likelihood estimation of the parameters of the gamma distribution and their bias, *Technometrics* 11 (4) (1969) 683–690.
- [24] H. Cao, V. Govindaraju, Handwritten carbon form preprocessing based on Markov random field, in: *Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'07)*, 2007, pp. 1–7.
- [25] R. Milewski, V. Govindaraju, Extraction of handwritten text from carbon copy medical form images, in: *Document Analysis Systems*, 2006, pp. 106–116.

About the Author—HUAIGU CAO received his Bachelor's and Master's degrees from the Electronic Engineering Department in Tsinghua University, Beijing, China in 2001 and 2004, respectively. He received his Ph.D. degree from the Department of Computer Science and Engineering (CSE), University at Buffalo, State University of New York (SUNY) in 2008. His research of interest includes image processing and pattern recognition. In 2008, Dr. Cao became a scientist with BBN Technologies. This paper is part of his research work finished in University at Buffalo.

About the Author—ANURAG BHARDWAJ obtained his Bachelor's degree in Computer Engineering from National Institute of Technology, Kurukshetra. He obtained his Master's degree from the Department of Computer Science and Engineering, University at Buffalo, State University of New York (SUNY). Now he is a doctoral student under the supervision of Dr. Venu Govindaraju at the Center for Unified Biometrics and Sensors (CUBS), CSE, University at Buffalo, State University of New York (SUNY). His research interests are Machine Learning, Information Retrieval and Document Analysis.

About the Author—VENU GOVINDARAJU is a Professor of Computer Science and Engineering at the University at Buffalo, State University of New York (SUNY). He received his B.Tech. (Honors) from the Indian Institute of Technology (IIT), Kharagpur, India in 1986, and his Ph.D. from UB in 1992. In a research career spanning over 20 years, Dr. Govindaraju has made significant contributions to the areas of pattern recognition such as document analysis and biometrics. He has authored more than 245 scientific papers including 45 journal papers and 20 book chapters. He has been the PI/Co-PI of projects funded by government and industry for about 50 million dollars in the last 15 years. Dr. Govindaraju is a Fellow of the IEEE (Institute of Electrical and Electronics Engineers) and a Fellow of the IAPR (International Association of Pattern Recognition).