

Mapping Transcripts to Handwritten Text

Chen Huang, Sargur N. Srihari

► **To cite this version:**

Chen Huang, Sargur N. Srihari. Mapping Transcripts to Handwritten Text. Guy Lorette. Tenth International Workshop on Frontiers in Handwriting Recognition, Oct 2006, La Baule (France), Suvisoft, 2006. <inria-00112763>

HAL Id: inria-00112763

<https://hal.inria.fr/inria-00112763>

Submitted on 9 Nov 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Mapping Transcripts to Handwritten Text

Chen Huang and Sargur N. Srihari

CEDAR, Department of Computer Science and Engineering
State University of New York at Buffalo
E-Mail: {chuang5, srihari}@cedar.buffalo.edu

Abstract

In the analysis and recognition of handwriting, a useful first task is to assign ground truth for words in the writing. Such an assignment is useful for various subsequent machine learning tasks for performing automatic recognition, writer verification, etc. Since automatic word segmentation and recognition can be error prone, an intermediate approach is to use a text file that is a transcription of the handwriting image for performing ground truth assignment. This paper describes an algorithm for finding the best word level alignment between the transcript and the handwriting image. The algorithm is useful in tasks such as: (i) extracting words and characters as characteristic elements in writer verification and identification tasks; (ii) creating a large ground-truthed dataset for handwriting document analysis (in word and even character levels); (iii) indexing a collection of handwritten materials for document retrieval, such as for historical manuscripts. The algorithm achieves an 84.7% accuracy in aligning words on whole images when evaluated on 20 pages from a handwriting database created for forensic document examination studies.

Keywords: Transcript mapping, word segmentation, word recognition.

1. Introduction

Handwriting analysis and recognition have been actively studied in the past thirty years. However, unconstrained off-line handwriting recognition and retrieval still remains a challenging problem. Usually, word recognition algorithms suffer from two aspects. One is the segmentation error from the word segmentation process, especially for the cursive handwriting documents. The other is that the accuracy of recognition drops when increasing the size of the lexicon. For example, in [2], the accuracy of the word recognizer is 96.8%, 88.23% and 73.8% when the size of lexicon is 10, 100 and 1,000 respectively. However sometimes transcriptions are available for some handwritten documents. In this case, word recognition problem for an entire document becomes an alignment problem between the handwritten document image and its transcript.

Even with a transcription available, it does not mean the alignment is a trivial problem. First of all, errors are

produced in the processing of word segmentation for unconstrained handwriting documents. The errors include both oversegmentations (i.e. one word image was separated as two or more fragments) and undersegmentations (i.e. two or more word images were grouped together and returned as one word image). Thus the total number of segmented word images usually is not equal to the total number of the text word in the transcript. Therefore a simple linear alignment will certainly not work. Secondly, even with a correctly segmented word image, word recognition may still produce error, and the accuracy of recognition drops if a large vocabulary lexicon is given. (Normally a full page of handwriting document can easily contain a 150-200 words vocabulary.) For these reasons, we propose a recognition-based alignment algorithm to solve these two problems simultaneously, and the output of our algorithm is an optimal mapping between the document image and its corresponding transcript. In addition, while we try to find the word by word correspondences for entire document, we don't assume any line by line correspondence information between the document image and the transcript.

Such mapping algorithm may have many applications. It could be useful in many research and real applications of handwriting processing, such as writer verification and identification in forensic science, designing and evaluating handwriting recognition techniques with a large ground truthed database, and handwritten document retrieval in digital libraries. They are described as follows.

In the field of forensic document examination (FDE), writer verification is the task of determining whether two handwriting samples were written by the same writer or not. In contrast, the task of writer identification is to determine for a questioned document as to which individual, with known handwriting, it belongs to. Recent research in studying the individuality of handwriting [6][8][9], has shown the effectiveness of handwritten words and characters for writer identification and verification tasks. However, in the previous studies [8][9], all character and word images were manually extracted from a large number of handwritten documents and ground-truthed. Automatic word recognition (with a lexicon size of about 150) and automatic character recognition were experimented and the discriminative power of the resulting words and characters decreased significantly. However manually ex-

traction of word and character images requires a lot of effort and time. Therefore automatic word recognition with transcript mapping may become an intermediate approach.

With more and more handwritten materials being added to today’s digital libraries, handwritten document retrieval becomes another interesting and important topic [5][7]. The task of it is to search through a repository of scanned handwritten documents for those most similar to a given query writing which could be either a text or an image of word or phrase. With a transcription available, we can index the handwritten document images by transcript mapping. As a result, not only the text-to-image retrieval becomes a straightforward problem (i.e. performing the text retrieval first and then mapping to images), but also the image-to-image retrieval will become possible and its performance will be improved, since a good alignment algorithm can help to get more accurate word images.

In addition, in the research field of handwriting recognition and retrieval, a large ground-truthed dataset of handwriting words and characters is always desirable for designing and evaluating techniques and algorithms [10].

The remainder of this paper is organized as follows. Section 2 discusses some related works and how our approach differs. We then formally define the problem and describe the proposed algorithm in details in Section 3. Section 4 presents the experimental results. Section 5 concludes the paper.

2. Related Work

As mentioned above, there are two difficulties in general word recognition: errors from word segmentation and large vocabulary lexicon. Because of such reasons, Kornfield et al. [4] proposed an alternative approach. Instead of doing word recognition for each segmented word image, they treat the set of word images and the transcript as two time series and then use dynamic time warping (DTW) to align them.

Similarly, without performing word recognition explicitly for each word image, Rothfeder et al. [5] use a linear Hidden Markov model (HMM) to solve the alignment problem. The HMM was constructed as follows. All the word images were treated as the hidden variables, while the feature vectors, extracted from each of the word images, being modeled as observed variables. The HMM models the probability of generating (observing) the word images given the words. The Viterbi algorithm was used to decode the sequence of assignments to each of the word images. Both of the DTW and HMM methods were evaluated on a set of 70 pages of George Washington collection and an average accuracy of 60.5% and 72.8% were reported respectively.

While we agree that a perfect line and word segmentation is impossible for cursive handwriting documents, we believe that by reducing the size of the lexicon using the information provided by the transcript, segmentation problem and alignment problem can help with each other

and be solved simultaneously. Therefore the approach we proposed is a recognition-based alignment algorithm that optimally utilize the information from a document image and its corresponding transcript to get the best mapping results. A word recognizer, called WMR (see details in section 3.3, is used to perform the task of word recognition. And a dynamic programming algorithm is used to find the optimal alignment between two word strings: the first one is the truth from the transcription and the second one is the recognition results from the word recognition. Since WMR will generate multiple choices as the result of word recognition, therefore for each word image sequence, multiple hypotheses are formed as the second word string and the best one will be found through the dynamic programming algorithm with the highest alignment score.

3. Algorithm Description

3.1. Problem Definition

Before running our alignment algorithm, some pre-processing including image binarization, line separation and automatic word segmentation are performed. Again, errors may be produced in every step mentioned above. The results is a set of auto-segmented word images W , as shown in Figure 1, that is defined as follows:

$$W = \langle w_1, w_2, \dots, w_i, \dots, w_n \rangle \quad (1)$$

where n is the total number of word images, and w_i represents one word image segment. Normally we will have three types of situations for w_i . (i) It may contain just one word, such as w_1 and w_5 in the example. In this case word segmentation is correct. (ii) Or it may contain more than one word (i.e. undersegment error). For example, w_4 grouped “of” and “the” as one word segment. (iii) Or it may only contain a part of a word (i.e. oversegment error), such as w_2 and w_3 in the example, which should be combined together and mapped to only one word, “cosponsor”, in the transcript.

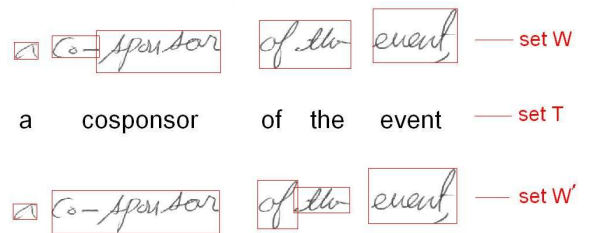


Figure 1. Examples for different sets in the problem definitions.

On the other hand, for a corresponding document image we have a truth transcript T , as shown in the second line in Figure 1, that is an ordered list of textual words, expressed as:

$$T = \langle t_1, t_2, \dots, t_j, \dots, t_m \rangle \quad (2)$$

where m is the total number of textual words in the transcript and t_j is one text word.

Since the proposed algorithm does not only align the transcript with the document image, but it also try to improve the word segmentation, so we also define the improved set of word images W' , as shown in the third line in Figure 1, which is supposed to be more closer to the “ideal” segmentation result, as follows:

$$W' = \langle w'_1, w'_2, \dots, w'_j, \dots, w'_m \rangle \quad (3)$$

Similar to the situations for W , (i) each w'_j may be exactly same as some w_i , which indicates a correct auto-segmentation; or (ii) it may be part of w_i , which indicates a fix of undersegmentation; or (iii) it may be a combination of w_i and w_{i+1} , which indicates a fix of oversegmentation.

In addition, we set the size of set W' also to be m , which is the same as the set T . That is because the goal of our alignment algorithm is to assign one improved word segment to each of the textual words in T , and in this way, we can find an optimal mapping between a sequence of truth words, T (i.e. the transcript), and another sequence of better segmented word images W' .

3.2. Algorithm Description

First of all, a diagram of our algorithm is shown in Figure 2.

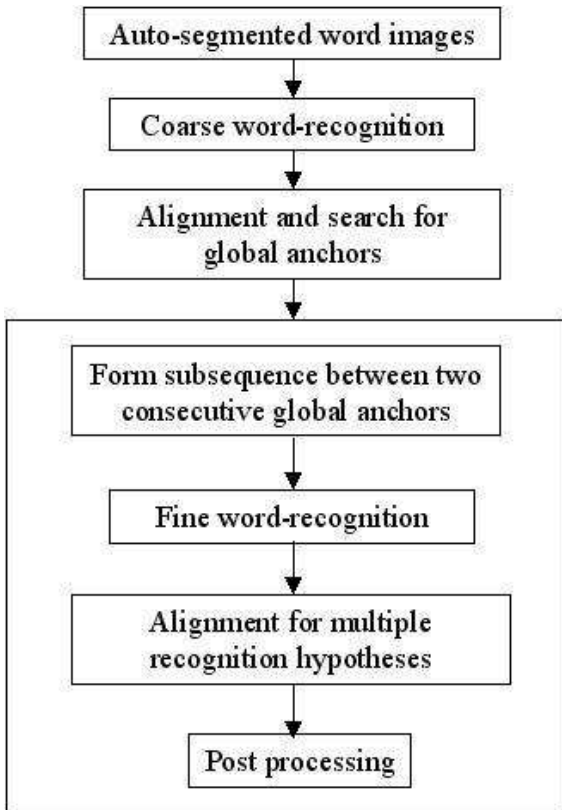


Figure 2. Diagram of the proposed algorithm.

The first step of our algorithm is to perform line sep-

aration and the word segmentation automatically. In our current system, a connected component based clustering algorithm is used to perform the line separation. Then a neural network based word segmentation algorithm is performed for every segmented line image. See [3] for the details of line segmentation and word segmentation algorithms.

A small modification was made at the word segmentation step. We add one constraint on the total number of auto-segmented word images. Because from the transcript we already know that the total number of textual words is m , we take advantage of this information and require that the total number of segmented word images n should be bounded by

$$m(1 - 0.15) \leq n \leq m(1 + 0.15) \quad (4)$$

After get all the auto-segmented word images, we perform a coarse word recognition for all the word images and then perform a coarse alignment on the entire document. The goal of this step is to seek for a set of global anchors. That is, for each auto-segmented word image w_j , we generate a lexicon based on its position information (i.e. the index number “ j ”), and then do word recognition for this word image. Here the size of the lexicon is chosen to be less than or equal to 20 (depending on the total number of words), which is a tradeoff between lexicon coverage and the recognition accuracy. The word recognizer will return an ordered list of lexicon words ranked in descending order of the dissimilarity value (i.e. distance measure). The details of the word recognizer will be discussed in the next section. In this step, only the top 1 choice of the returned word list will be considered as the truth of the word image.

After the coarse word recognition, each word image has been assigned a text word. Then a coarse alignment will be performed on the entire document in order to find a set of global anchors. Here the alignment problem is solved by a dynamic programming algorithm (see Section 3.4). A word image will be chosen as a global anchor if the following conditions are both satisfied: (i) this word is in the longest common subsequence that is generated from the dynamic programming algorithm, (ii) its associated distance value from the recognition result should be small (in our current system, it should be less than a trained threshold value).

After we get a set of global anchors, the next step is to segment the entire document into several subsequences. For every two consecutive global anchors, w_i and w_j , the set of all the word images in between is just a subsequence of the entire document, and we call it W_{ij} . Then each of such subsequence can be treated as a shorter document.

For each such subsequence, we perform a fine word recognition for each word image with a smaller lexicon (size of the lexicon is about 10). But this time, we may not only keep the top 1 word from the returned list, but may also keep the top 2 or top 3 as the possible candidates. The criteria for whether keeping more candidates are as follows. We choose top 2 only if the distance value of top-1

is large and the difference between the distance value of top-1 and top-2 is small (both criteria use some threshold values that were estimated using the training data). Therefore multiple hypotheses for the recognition string could be generated by choosing different choice, if they are available. After that, the same dynamic programming algorithm is used to find the optimal alignment between the truth string and each of the recognition string. Finally the hypothesis with the highest alignment score is selected.

A post processing will be performed after each subsequence gets a best alignment from all the hypotheses. It looks through the actual alignments and try to fix any mismatching alignment caused by segmentation errors. The details of post processing will be discussed in section 3.5.

3.3. Word-model Word Recognition

Word-model Word Recognition (WMR) [2] takes as inputs a word image and a lexicon, and computes a dissimilarity score (distance) between each lexicon word and the word image. And then the lexicon words are ranked and returned with the top 1 choice as the best match between the lexicon and the word image.

To match the word image against a lexicon, WMR involves three major phases, i.e., segmentation, feature extraction and matching. The segmentation phase separates a word image into smaller pieces called segments. Each segment represents a character or a sub-character (i.e. a part of a character). During the phase of feature extraction, 74 chain code based features are extracted from all possible combinations of 1-4 consecutive segments (called as super-segments). A super-segment corresponds to a single character in a word of the lexicon.

Given a lexicon word, the matching phase uses a dynamic programming algorithm to match features of the super-segments with the “ideal” features (obtained in the training procedure of WMR) of characters from the lexicon word, and computes a distance as the matching score. The matching phase is repeated for all lexicon words. As a result, the output of WMR consists of a list of lexicon entries ranked in descending order of their matching score values.

While the matching phase determines the segmentation points between segments that correspond to characters in a lexicon word, WMR can also be used to segment character images from a word image if a single true lexicon word is presented. Further more, it will be used in the post processing to fix some of the undersegmentation errors, i.e., segment an undersegmented word image into two or more word images.

3.4. Word String Alignment

String alignment problem is also a common problem in many research areas, such as in bioinformatics [1]. Here, given two textual sequences P and Q : one is from the transcription (the truth) and the other is formed by the results from word recognizer, we have a similar alignment

problem. The difference is that the elements of the string are no longer characters but words and the “string” here is really a sentence (i.e. a sequence of words). We design our alignment algorithm as follows.

Similar to the algorithm for string matching, we need to introduce a special word “-”, which represent the insertion of an empty word (or a gap). Given two strings P and Q , with $|P| = n$ and $|Q| = m$, in order to compute an optimal alignment of P and Q , we first define a *score function* as follows.

If p_i and q_j are each a single word or an empty word, then $\sigma(p_i, q_j)$ denotes the score of aligning p_i and q_j . In our problem, we define $\sigma(p_i, q_j)$ as

$$\sigma(p_i, q_j) = \begin{cases} 1 + c_j & \text{if } p_i = q_j \\ -1 & \text{otherwise} \end{cases} \quad (5)$$

where c_j is a confidence value for q_j and $0 \leq c_j \leq 1$. This confidence value is computed from the distance measure associated with each choice returned by the word recognizer.

Then we define $V(i, j)$ as the value of an optimal alignment of the string $\langle p_1, \dots, p_i \rangle$ and $\langle q_1, \dots, q_j \rangle$. The value of an optimal alignment of P and Q is then $V(n, m)$. And it's solved by using the following recurrence formula:

$$V(i, j) = \max \left\{ \begin{array}{l} V(i-1, j-1) + \sigma(p_i, q_j), \\ V(i-1, j) + \sigma(p_i, -), \\ V(i, j-1) + \sigma(-, q_j) \end{array} \right\} \quad (6)$$

After we get the score of an optimal alignment, the actual alignments can be recovered by retracing the dynamic programming steps back from the $V(n, m)$ entry.

3.5. Post Processing

The goal of the post processing is to improve the alignment results by detecting and fixing the potential segmentation errors. As we mentioned before, word segmentation usually has two types of errors: undersegmentation and oversegmentation.

Normally when undersegmentation happens, there will be an empty word in results sequence. As shown in Figure 3. In this case, usually it should be another mismatch in its neighbor (either before it or after it). So we can try to combine the missed word lexicon with the mismatched one to form a new word and add it into the current lexicon, and do the word recognition again using the new lexicon, if the new word is recognized as the top 1 choice, then this word image will be segmented into characters and then grouped as two words based on the truth words, and then a cut will be made at the boundary of these two word images. And two text words are assigned to them respectively.

In the case of oversegmentation, usually there will be a space in the truth sequence. As shown in Figure 4. In this case, there are two possibilities. One is that the corresponding image piece is not a word image and is even not

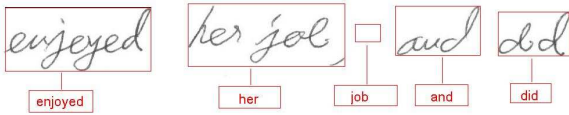


Figure 3. An example shows an undersegmentation error.

a part of word image. It could be a punctuation, such as “?” and “!”. The other is that it is a part of word. We can distinguish these two cases by doing the following. We attached it to its neighbors (to the end of the word before it, or to the front of the word after it), then we do word recognition for the new word image using the same lexicon again. If the returned text word of the top-1 choice is matched or its distance value is lower than the previous one, then we keep the new combined image. (we may just fix a oversegment) Or if the returned text is not matched or its distance value is greater than the previous one, then we leave that piece of image out.

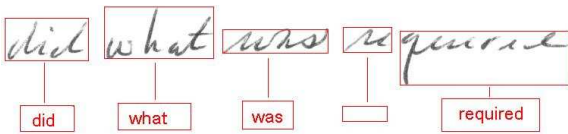


Figure 4. An example shows an oversegmentation error.

4. Experimental results and analysis

4.1. Dataset

The dataset used for experiment contains 20 pages (3120 words) handwritten document, which is a small subset chosen from a large dataset created for forensic document examination studies [6]. The content of the document is so called “CEDAR letter”, which was designed to contain 156 words including all characters (letters and numerals), punctuations and distinctive letter and numeral combinations (ff, tt, oo, 00). The vocabulary size is 124. That is, 32 out of 156 words are duplicate words, and most of them are the stop words, such as “the”, “she” and etc. About 1,500 individuals copied the CEDAR letter three times each in his/her most natural handwriting using plain unlined sheets, and a medium black ball-point pen. The samples were scanned using 300 dpi resolution and 8-bit grayscale. Figure 5 shows a sample image and the content of the CEDAR letter.

4.2. Experimental results

Since the goal of our algorithm is to assign an optimal word image segment for each text word from the transcript, therefore a mapping is evaluated as correct if the corresponding word image contains the exact word or the major part of the word. Then the accuracy is the total number of correct alignments divided by the total number of words in the transcript.

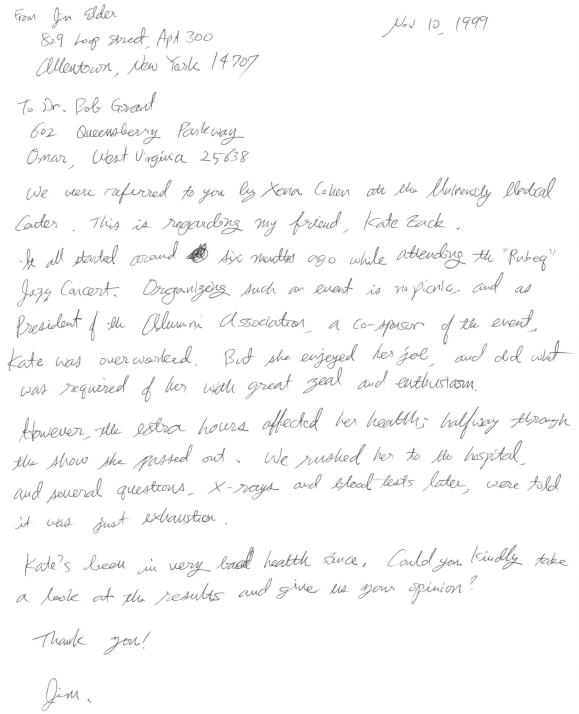


Figure 5. A handwriting sample from CEDAR letter dataset.

In order to show the improvement on word segmentation, we evaluated the performance for two cases: before the post processing and after the post processing. Before the post processing, the accuracy of the alignment is 78.3% (2443 words out of 3120 were alignmented correctly). After post processing, the accuracy is 84.7% (2643 words out of 3120 were alignmented correctly). This performance shows an improvement on some of our previous studies [7][10]. When compared with the accuracy of 60.5% reported in [4] and 72.8% in [5], our performance still shows the effectiveness of the proposed algorithm, although their experiments were performed on a larger set of historical documents, which are usually considered to be more difficult. The way they evaluated their alignment performance is also a slightly different than ours. Because their algorithms do not have any changes on automatic word segmentation, instead, in the case of oversegmentation, they assign the same text word to all its fragment images, while in the case of undersegmentation, they assign several corresponding text words to the undersegmented image.

5. Conclusion and Discussion

Alignment transcript to handwritten document is useful in many research or practical applications. We design a recognition-based alignment algorithm to solve this problem. In our algorithm, word recognition is performed based on a small size lexicon. The lexicon size is reduced by optimally utilizing the information provided by the transcript. The recognition results are aligned using

a dynamic programming algorithm. Multiple hypotheses of recognition results are generated for each subsequence separated by global anchors. The proposed algorithm also improve the word segmentation performance while doing the alignment. The high accuracy of the alignment is an indication of the effectiveness of the proposed method.

Currently we are performing some more extensive experiments, both on a larger dataset and on some different dataset (such as historical manuscript). Improving the line separation and removal of punctuation in word segmentation are also considered to be part of our future works.

References

- [1] R. Durbin, S. Eddy, A. Krogh and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 2001.
- [2] G. Kim and V. Govindaraju, "A lexicon driven approach to handwritten word recognition for real-time applications", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4), April 1997, pp 366-379.
- [3] G. Kim, V. Govindaraju and S. N. Srihari, "An architecture for handwritten text recognition systems", *International Journal on Document Analysis and Recognition*, 2(1), 1999, pp 37-44.
- [4] E. M. Kornfield, R. Manmatha and J. Allan, "Text alignment with handwritten documents", *Proceedings of the 1st International Workshop on Document Image Analysis for Libraries (DIAL)*, 2004, pp 195-209.
- [5] J. Rothfeder, R. Manmatha and T. M. Rath, "Aligning transcripts to automatically segmented handwritten manuscripts", *Proceedings of the 7th IAPR Workshop on Document Analysis Systems*, Nelson, New Zealand, February 2006, pp 84-95.
- [6] S. N. Srihari, S. Cha, H. Arora and S. Lee, "Individuality of handwriting", *Journal of Forensic Sciences*, 47(4), July 2002, pp 1-17.
- [7] C. I. Tomai, B. Zhang and V. Govindaraju, "Transcript mapping for historic handwritten document images", *Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition*, Niagara-on-the-Lake, Ontario, Canada, August 2002, pp 413-418.
- [8] B. Zhang, S. N. Srihari and S. Lee, "Individuality of handwritten characters", *Proceedings of the 7th International Conference on Document Analysis and Recognition*, Edinburgh, Scotland, August 3-6, 2003, pp 1086-1090.
- [9] B. Zhang and S. N. Srihari, "Analysis of handwriting individuality using handwritten words", *Proceedings of the 7th International Conference on Document Analysis and Recognition*, Edinburgh, Scotland, August 3-6, 2003, pp 1142-1146.
- [10] B. Zhang, C. I. Tomai, S. N. Srihari and V. Govindaraju, "Construction of handwriting databases using transcript-based mapping", *Proceedings of the 1st International Workshop on Document Image Analysis for Libraries (DIAL)*, 2004, pp 288-298.