2014 14th International Conference on Frontiers in Handwriting Recognition

A Novel Transcript Mapping Technique for Handwritten Document Images

Nikolaos Stamatopoulos, Basilis Gatos and Georgios Louloudis
Computational Intelligence Laboratory, Institute of Informatics and Telecommunications
National Center for Scientific Research "Demokritos"
GR-153 10 Agia Paraskevi, Athens, Greece
{nstam, bgat, louloud}@iit.demokritos.gr

Abstract— *Transcript mapping refers to the process of aligning meaningful units of a handwritten document image (e.g. text lines, words, characters) with the corresponding transcription information. It has many applications such as (i) fast generation of ground truth at different granularity levels and (ii) indexing handwritten collections for document retrieval. In this paper, a novel transcript mapping technique is proposed which is guided by the number of words as well as the characters per word of a text line. The proposed method combines the results of a local and a global approach using a scoring algorithm. The efficiency of the proposed method is demonstrated by experimentation conducted on a known, publicly available dataset, achieving word level alignment accuracy of 99.48%.*

Keywords- transcript mapping; word segmentation

I. INTRODUCTION

Over the last years, mass digitization has become one of the most prominent issues in the library world. Considerable effort has been invested in creating digital libraries of the heritage of many societies around the world. However, there is still a growing need for systems and tools that will automatically make the content of handwritten document images browsable and searchable. Furthermore, when the transcription is available, a correlation between the correct text and the corresponding image area in text line and word level (see Fig. 1) can be particularly useful for scientists in the fields of literature and history.

Transcript mapping techniques are used in order to align the correct text information to a segmentation result produced automatically. The result of transcript mapping also permits a fast generation of benchmarking/training datasets for text line or word recognition since the completely manual creation is a difficult and time-consuming task. After the transcript mapping procedure, a minimum user involvement for the correction of segmentation errors is necessary in order to produce the final ground truth.

In this paper, a novel transcript mapping technique is proposed which assumes the existence of line breaks and is guided by the number of words as well as the characters per word of a text line. The proposed method combines the results of a local and a global approach using a scoring algorithm. The local approach is a modification of our previous method [1]. A post-processing step has been added which takes into account the number of characters of each word in a text line. Concerning the global approach, the optimal segmentation result among several segmentation hypotheses is produced by minimizing a suitable cost function. Finally, a selection is made based on a scoring algorithm applied on the previously calculated results.

Figure 1. Screenshot of a prototype interface displaying correlation of the transcription with the corresponding image area in text line and word level.

The efficiency of the proposed technique is demonstrated by experimentation conducted on the test set of the ICDAR2009 Handwriting Segmentation Contest [2] achieving word level alignment accuracy of 99.48%. The flowchart of the proposed technique for the creation of document image segmentation at word level that includes text-image alignment is demonstrated in Fig. 2.

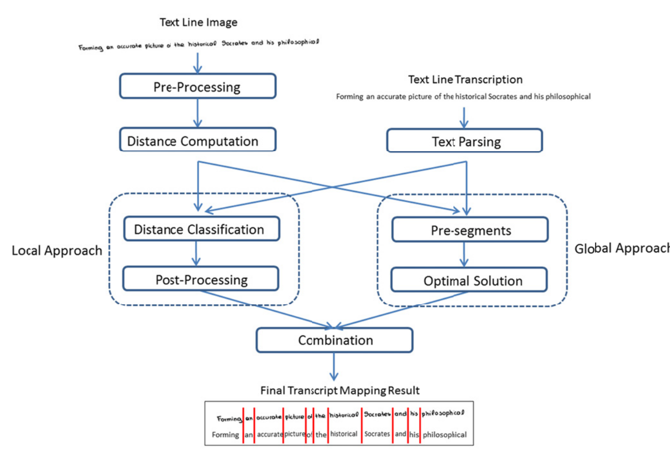

Figure 2. Flowchart of the proposed technique.

2167-6445/14 $31.00 © 2014 IEEE
DOI 10.1109/ICFHR.2014.15

The remainder of the paper is organized as follows. In Section II, related work is presented. In Section III, the proposed method is detailed while experimental results are discussed in Section IV. Finally, conclusions are drawn in Section V.

II. Related Work

Transcript mapping methods fall broadly into two main categories according to the algorithm which is used for the alignment. The first category contains methods which use recognition models (supervised scenario) whereas the second category includes methods which work in an unsupervised environment (no training is involved) using image features and transcription information. The analysis is focused on word level alignment starting from correct text lines as input.

Concerning the first category, Zimmermann and Bunke [3] present an automatic segmentation scheme for cursive handwritten text lines using transcriptions of text lines and an HMM based recognition system. The segmentation scheme consists of two steps. In the first step, the Viterbi decoder is used in forced alignment mode using a normalized text line as input, in which optimal word boundaries can be computed. The second step uses these word boundaries to assign the connected components of the normalized line to individual words. The authors report a 98% word segmentation rate on the IAM database. An extension of this work is presented in [4] where the authors tackle mismatches between the words presented in the image and the transcription (e.g. abbreviations, capitalization). The method also aligns complete document pages instead of single text lines and achieves considerable alignment accuracy even with weakly trained character models. In a similar direction, Toselli et al. [5] propose an alignment method based on the Viterbi algorithm to find mappings between word images of a given handwritten document and their respective words on the transcription. This method takes advantage of the implicit alignment made by Viterbi decoding used in text recognition with HMMs. In [6], Rothfeder et al. use a linear HMM to solve the alignment problem without performing word recognition explicitly for each word image. All the word images are treated as hidden variables, while the feature vectors which are extracted from the word images, are modeled as observed variables. The Viterbi algorithm is used to decode the sequence of assignments to each of the word images. The authors evaluate the method on a set of 70 pages of George Washington collection and an average accuracy of 72.8% is reported. Tomai et al. [7] propose a method in order to limit the lexicon of a handwriting recognizer using the transcription. A ranked list of possible words from the lexicon is returned for each recognized word image. Several segmentation hypotheses of a line are suggested. Then, word mapping is defined using word recognition results by a dynamic programming algorithm that finds the best match. In a similar manner, Huang and Srihari [8] present a recognition-based alignment algorithm. A word recognizer generates multiple choices as a result. Then, dynamic programming is used to find the optimal alignment between two word strings: the first one is the truth from the transcription and the second one corresponds to the results of word recognition. The authors report 84.7% accuracy in aligning words on 20 pages of a handwritten database. Fei Yin et al. [9] present a method which formulates transcript mapping as an optimization problem, incorporating the geometric context of characters as well as character recognition models. The method is tested on Chinese handwriting pages and achieves alignment at character level.

The second category of methods includes the work of Kornfield et al. [10] which relies on dynamic time warping (DTW) and uses as input series the image locations from the segmentation step and the text words in transcription. This method does not require performing word recognition for each segmented word image. It was applied to the historical handwritten documents of Washington collection and achieved 60.5% accuracy when aligning full pages. Lorigo and Govindaraju [11] propose a transcript mapping method for handwritten Arabic documents. It is based on an extension of DTW that uses true distances when mapping multiple entries from one series to a single entry in the second series. In [12], Zinger et al. present a method for text-image alignment in the context of building a historical document retrieval system. The images of handwritten lines are automatically segmented from the scanned pages of historical documents and then manually transcribed. Alignment on word level is based on the longest spaces between portions of handwriting. A cost function is defined considering the relative word length. The minimum value of the cost function defines the word boundary. Our previous technique [1] belongs to the group of methods that work in an unsupervised environment. For the alignment at word level, we adopted an algorithm which repeatedly separates the text line using the between-component gaps as threshold until the number of words created on the image equals the number of words of the transcription. An accuracy of 97.13% is achieved on the test set of the ICDAR2009 Handwriting Segmentation Contest [2].

It is clear that most state-of-the-art methods are of supervised nature leading to high performance but have the disadvantage of needing a training phase which makes necessary the existence of annotated data beforehand. In order to overcome this drawback, we propose an unsupervised method that uses both the number of words and the characters per word of a text line. Experimental results prove that the alignment accuracy (which is >99%) is comparable to accuracies of supervised methods without any need for training.

III. Proposed Method

We assume that the transcription includes the correct text line break information. For each text line, the transcription is first parsed and this information is used in a local and a global transcript mapping technique. Finally, a selection is made based on a scoring algorithm. All involved stages are detailed in this section.

A. *Text Parsing*

Before we proceed with the transcript mapping techniques, useful information from the transcription of a text line is extracted. By using a simple text parser, the number of words *NW* as well as the number of characters NC_i for each word *i* are calculated.

B. Local Approach

The local approach is a modification of the method described in [1] which is based on a distance classification technique constrained by the number of words for each text line. A post-processing step has been added which takes into account the number of characters of each word in the text line. It includes the following steps:

1. Pre-processing

The pre-processing step concerns the correction of the skew angle using the technique proposed in [13] as well as the dominant slant angle of the text line image based on [14]. Fig. 3 shows the resulting images after applying skew and slat correction.

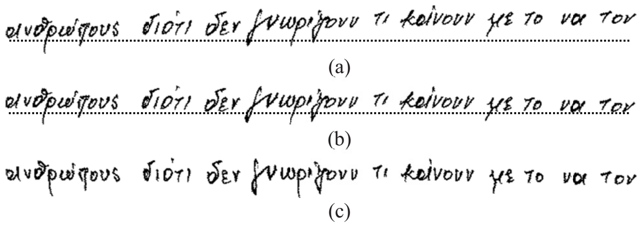

Figure 3. Pre-processing procedure: (a) original text line image, (b) image after skew correction and (c) image after slant correction.

2. Distance Computation

This step deals with the computation of the distances of adjacent components in the text line image [1]. The computation of the distance metric is considered on adjacent overlapped components (OCs). An OC is defined as a set of connected components whose projection profiles overlap in the vertical direction. We define as distance of two adjacent OCs their Euclidean distance which is the minimum among the Euclidean distances of all pairs of points of the two adjacent OCs.

3. Distance Classification

At this step, every distance is classified as inter-word or intra-word. For this classification we use a local threshold for every text line of the image. All distances above this threshold are considered as inter-word distances whereas all distances below this threshold are considered as intra-word distances. We select as threshold the largest distance which produces equal or larger number of words from the actual number of words *NW*.

4. Post-processing

Once the words have been detected we proceed to a post-processing step in order to correct possible segmentation errors using the number of characters of each word. We either split or merge a detected word when its width deviates from a statistical estimation based on the number of the characters of the word.

For each word i of the transcription, moving from left to right over the text line, we calculate the cost function $\mathcal{F}_i$ using the following equation:

$$\mathcal{F}_i = (NC_i * AW) - W_i \quad (1)$$

where W_i is the width in pixels of the corresponding detected word in the image and AW the average character width in the text line defined as follows:

$$AW = \frac{\sum_{j=1}^{ND} W_j}{\sum_{l=1}^{NW} NC_l} \quad (2)$$

where ND is the number of words detected at the previous step.

The word segmentation result is modified according to the cost function $\mathcal{F}_i$ in order to minimize it. There are three different cases based on a threshold T_i :

- $-T_i < \mathcal{F}_i < T_i$: The word i has been detected correctly.
- $\mathcal{F}_i > T_i$: The word i has to be merged with the following detected word only if this merging reduces the cost function. According to this criterion, we merge two words when the width of the detected word is smaller than the expected width as it is defined from the number of characters.
- $\mathcal{F}_i < -T_i$: The word i has to be split into two words since its width is larger than the expected. The word is split into all possible ways using the distances of adjacent overlapped components and we select the one which minimizes the cost function of the left word.

The threshold T_i depends on the average character width. In the case of large words, we modify the segmentation result when the cost function deviates three times the average character width. Otherwise, the threshold T_i depends also on the number of characters of the word and it is calculated using the following equation:

$$T_i = \begin{cases} 3 * AW & if\ NC_i > 5 \\ (NC_i/2) * AW & if\ NC_i \leq 5 \end{cases} \quad (3)$$

In Fig. 4 we present an example of a word segmentation result after applying the post-processing step. The values of the cost function $\mathcal{F}_i$ and the threshold T_i before and after applying changes are presented.

C. Multiple Hypothesis Global Approach

In this section, we introduce a novel transcript mapping approach that is based, at a first step, on segmenting every text line in a set of pre-segments and, at a next step, on the creation of a word segmentation hypothesis set that leads to the optimal segmentation solution by minimizing a suitable cost function.

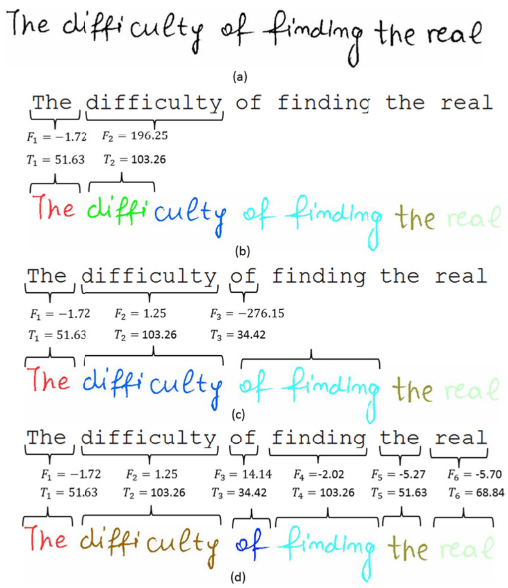

Figure 4. An example of the local approach after applying the post-processing step using the cost function $\mathcal{F}_i$ and $AW = 34.42$: (a) original text line image; (b) initial segmentation result where the second word needs merging ($\mathcal{F}_2 > T_2$); (c) after merging, the third word needs spiltting ($\mathcal{F}_3 < -T_3$) and (d) the final segmentation result.

1. Calculation of pre-segments

The pre-segments are calculated following the first two steps of the local approach (see Sections III.B.1 and III.B.2) and then applying a distance threshold T_{pre} that results in a maximum of $NW+n$ words, where NW is the number of words of current text line and n is a parameter related to the desired over-segmentation flavor of the result (for our experiments n is set to 2) (see Fig. 5c). T_{pre} is calculated by checking all segmentation results starting by applying a distance threshold T_{pre} =0 (this results to a number of words equal to the number of overlapped components – see Section III.B.2) and then increasing this value until the resulting number of words is less or equal to $NW+n$.

2. Finding the optimal solution

At this step, we proceed to consecutively merging all neighboring pre-segments in order to have the number of estimated words equal to NW. This can be accomplished by an exhaustive search of all possible combinations of pre-segment merging.

In order to select the optimal solution from all p possible results, we define the following cost function that needs to be minimized:

$$C_k = \sum_{i=1}^{NW}\sum_{j=1}^{NW}\left|\frac{W_i}{W_j} - \frac{NC_i}{NC_j}\right| \quad (4)$$

where k is current segmentation result (k=1...p), W_i the width of the i-th detected word and NC_i the number of characters of the i-th word of current text line (see Fig. 5d). The rationale behind the proposed cost function is that the ratio of the widths of any detected words pair must be approximately equal to the ratio of the number of characters of the corresponding words pair only when the word segmentation result is correct.

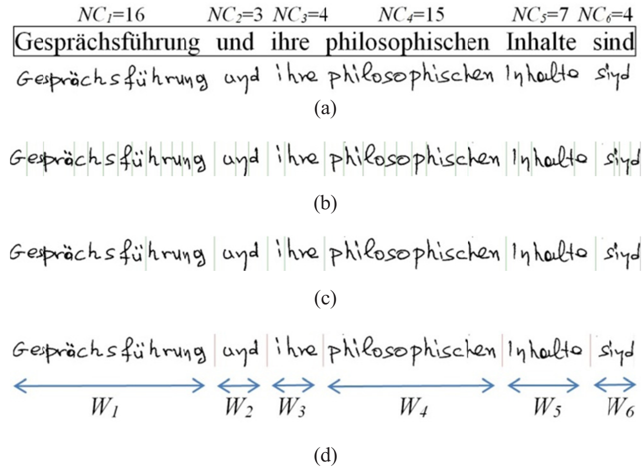

Figure 5. Example of the multiple hypothesis global transcript mapping approach: (a) Original text line image and the corresponding transcription; (b) the existing overlapped components; (c) the pre-segments calculated during the first step of the method; (d) the optimal segmentation solution provided by the method.

D. Selection of Segmentation Result

After the calculation of the two segmentation results for each text line using the local and global approaches, respectively, a final selection is made based on a scoring algorithm applied on both results. The segmentation result with the lowest score is considered as final. In the case of equal scores, one of the two results is selected. The scoring algorithm is based on the ranking of both text and image with respect to the word width (number of characters per word in the case of text and width per word in the case of image). A more detailed description of the score calculation method is provided in this section.

1. Text Ranking

The number of characters per word is used for the calculation of the text ranking. In more detail, the words are sorted in descending order with respect to their number of characters. The word with the largest number of characters is considered to have rank 1, the word with the second largest number of characters is considered to have rank 2, etc. Fig. 6 presents the ranking of the words for a text line.

Text	Socrates	was	a	Classical	Greek	philosopher.	Credited	as	one	of	the
Characters	8	3	1	9	5	12	8	2	3	2	3
Ranking	3	6	11	2	5	1	3	9	6	9	6

Figure 6. Text ranking. The ranking value (last row) is based on the number of characters of each word (second row).

2. Image Ranking

A similar procedure is followed for image ranking. However, instead of sorting the words using the number of characters we use the width of the word images (counted in pixels). The ranking is produced after sorting the widths in descending order (see Fig. 7).

Image											
Width	208	92	32	225	134	268	204	54	80	44	77
Ranking	3	6	11	2	5	1	4	9	7	10	8

Figure 7. Image ranking. The ranking value (last row) is based on the width of the word images (second row).

3. *Image Ranking* adjustment

A common outcome of the text ranking step is the assignment of the same ranking value in two or more words. The main reason for this is that, often, words appearing in a text line have the same number of characters (see for example columns 1, 7 of Fig. 6). This situation rarely appears at the image ranking step. For better comparison of the two rankings, an adjustment of the image ranking is accomplished using the text ranking. After sorting both the text and image rankings we assign the text ranking id to the image ranking id (see Fig. 8). This adjustment is used to update the image ranking values (see Fig. 9).

Sorted Text Ranking	1	2	3	3	5	6	6	6	9	9	11
Sorted Image Ranking	1	2	3	4	5	6	7	8	9	10	11
Adjusted Image Ranking	1	2	3	3	5	6	6	6	9	9	11

Figure 8. Adjusted image ranking. The adjusted image ranking (last row) is created by assigning the values of the sorted text ranking (first row) to the values of the sorted image ranking (second row).

Image											
Width	208	92	32	225	134	268	204	54	80	44	77
Ranking	3	6	11	2	5	1	3	9	6	9	6

Figure 9. Image ranking after adjustment. The final image ranking (last row) after adjustment. Changed ranking values are underlined.

4. *Score Calculation Ranking* adjustment

The final step involves the comparison of the previously calculated rankings (text and image level) in order to produce a final score ***S***. The initial value of ***S*** is set to zero. In a left to right order starting from leftmost column, the ranking values are compared. Let *T[i]*, *Im[i]* correspond to the ranking values of the text and the image for the i^{th} word, respectively (*i=1...NW*) and *U[j]* corresponds to the set of unique ranking values sorted in ascending order ($j=1...M \leq NW$). *U[j]* is defined in order to provide information about adjacent ranking values. The main idea of scoring is that no score is added at the cases where the ranking values are the same (Fig. 10) as well as when two adjacent ranking values change position (Fig. 11). In all other cases, the absolute difference of the ranking values is added to the total score ***S*** (Fig. 12). The following algorithm describes the calculation of the final score.

```
S = 0;
for (i=1;i<=NW;i++)
{
  if (T[i] == Im[i]) //if ranking values are the same
    S = S + 0; //Do nothing
  else
  {
   int rank_left = find_adjacent_value_left(T[i]);//using the unique list U[j]
   int rank_right = find_adjacent_value_right(T[i]);//using the unique list U[j]
   if (Im[i] == rank_left  OR Im[i] == rank_right)
     bool found  = Search_the_ranking_list_for_a_reverse_tuple();
     //(i.e. find a tuple T[j] - Im[j] where T[j] = Im[i] and T[i] = Im [j])
     if (found)
       S = S + 0;
     else
       S = S + abs(T[i]-Im[i]);
   }
}
return S;
```

T[i]	3	6	11	2	5	1	3	9	6	9	6
Im[i]	3	6	11	2	5	1	3	9	6	9	6
Column_Score	0	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	0	0	0	0

Figure 10. Score calculation example where all ranking values are equal thus resulting in zero column score as well as zero total score.

T[i]	5	7	1	4	11	7	1	5	1	7	7
Im[i]	5	7	1	1	11	7	4	5	1	7	7
Column_Score	0	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	0	0	0	0

Figure 11. Score calculation example where a tuple of adjacent ranking values (first green column – values 4,1) exists in reverse order in another column (second green column). Both columns get a zero column score resulting in zero total score.

T[i]	7	1	3	7	2	5	5	3	7	7	1
Im[i]	7	1	3	7	7	5	3	5	2	7	1
Column_Score	0	0	0	0	5	0	0	0	5	0	0
S	0	0	0	0	5	5	5	5	10	10	10

Figure 12. Score calculation example where an image ranking value is not adjacent to the text ranking value (first red column – 7 is not adjacent to 2) resulting in a column score of 5. A similar condition is met in column 9 (second red column). The total score is 10.

IV. Experimental Results

The proposed transcript mapping method was evaluated on the test set of the ICDAR2009 Handwriting Segmentation Contest [2]. The test set consists of 200 document images written in several languages (English, French, German and Greek) that contain 29717 words.

The performance evaluation method is based on counting the number of matches between the words detected by the algorithm and the words in the ground truth. A word pair is considered as a one-to-one (*o2o*) match only if the matching score is equal to or above the evaluator's acceptance threshold $T_a = 90\%$. Let *N* be the count of ground-truth words and *M* be the count of result words. Detection rate (*DR*) and recognition accuracy (*RA*) are defined as follows:

$$DR = \frac{o2o}{N}, RA = \frac{o2o}{M} \quad (5)$$

The performance metric *FM* is extracted as follows:

$$FM = \frac{2 * DR * RA}{DR + RA} \quad (6)$$

We compared the proposed method with our previous technique [1] as well as the winner of the ICDAR2009

Handwriting Segmentation Contest [2], [15] (ILSP-LWSeg-09). Moreover, the evaluation results of the local and global approaches before their combination are presented. Table I illustrates the overall comparative evaluation results.

TABLE I. COMPARATIVE EVALUATION RESULTS

Method	*M*	*o2o*	*DR (%)*	*RA (%)*	*FM (%)*
ILSP-LWSeg-09	29962	28279	95.16	94.38	94.77
Previous Method	29673	28845	97.06	97.21	97.13
Local Approach	29717	29370	98.83	98.83	98.83
Global Approach	29717	29499	99.26	99.26	99.26
Combination	**29717**	**29563**	**99.48**	**99.48**	**99.48**

As the evaluation results indicate, the local and global approaches, using as additional information the number of characters of each word, outperform our previous method [1] which is based only on the number of words for each text line. Furthermore, the proposed method, after the combination of the two methods, achieved the best results with *FM* = 99.48%. This means that it fails to correctly detect only 154 words out of 29717. As a result, only a very small number of segmentation results needs correction in order to produce the final word segmentation ground truth. Fig. 13 depicts representative examples of the two most common cases in which the method fails due to (i) the presence of punctuation marks which are merged with the following word (Fig 13(b)) or (ii) overlapped words which cannot be split (Fig 13(d)).

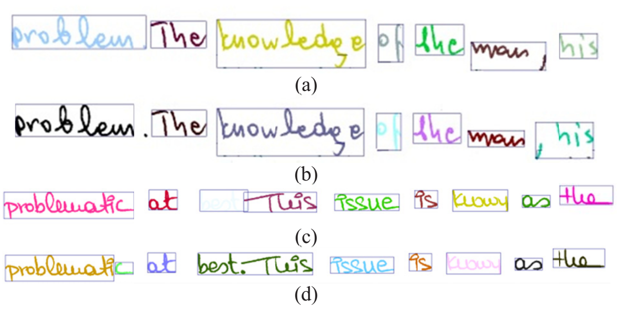

Figure 13. Representative erros: (a),(c) segmentation ground truth (b),(d) word segmentation result produced by the proposed method.

V. CONCLUSIONS

A novel transcript mapping method for aligning the correct text information to the image at word level is presented. It works in an unsupervised environment using image features and transcription information namely the number of words and characters per text line. Two new approaches are introduced, a local approach which is an extension of a previous method [1] and a global approach in which optimal segmentation result is selected among several segmentation hypotheses by minimizing a suitable cost function. A scoring algorithm is used to select between the two methods. Experimental results prove that the alignment accuracy is very high (99.48%) and comparable to accuracies of supervised methods without any need for training.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 600707 - tranScriptorium.

REFERENCES

[1] N. Stamatopoulos, G. Louloudis and B. Gatos, “Efficient Transcript Mapping to Ease the Creation of Document Image Segmentation Ground Truth with Text-Image Alignment”, 12th International Conference on Frontiers in Handwriting Recognition, pp. 226-231, Kolkata, India, 2010.

[2] B. Gatos, N. Stamatopoulos and G. Louloudis, “ICDAR2009 Handwriting Segmentation Contest”, 10th International Conference on Document Analysis and Recognition, pp. 1393-1397, Barcelona, Spain, 2009.

[3] M. Zimmermann and H. Bunke, “Automatic Segmentation of the IAM Off-line Database for Handwritten English Text”, 16th International Conference on Pattern Recognition, pp. 35-39, Quebec, Canada, 2002.

[4] A. Fischer, V. Frinken, A. Fornés and H. Bunke, “Transcription alignment of latin manuscripts using hidden markov models”, 1st Workshop on Historical Document Imaging and Processing, pp. 29–36, Beijing, China, 2011.

[5] A. Toselli, V. Romero and E. Vidal, “Viterbi based alignment between text images and their transcripts”, Workshop on Language Technology for Cultural Heritage Data, pp. 9-16, Prague, Czech Republic, 2007.

[6] J. Rothfeder, R. Manmatha and T.M. Rath, “Aligning Transcripts to Automatically Segmented Handwritten Manuscripts”, 7th International Workshop on Document Analysis Systems, pp. 84-95, Nelson, New Zealand, 2006.

[7] C. Tomai, B. Zhang and V. Govindaraju, “Transcript Mapping for Historic Handwritten Documents”, 8th International Workshop on Frontiers in Handwriting Recognition, pp. 413-418, Ontario, Canada, 2002.

[8] C. Huang and S.N. Srihari, “Mapping transcripts to handwritten text”, 10th International Workshop on Frontiers in Handwriting Recognition, pp. 15-20, La Baule, France, 2006.

[9] F. Yin, Q. Wang and C.L. Liu, “Transcript mapping for handwritten Chinese documents by integrating character recognition model and geometric context”, Pattern Recognition, vol. 46, no. 10, pp. 2807-2818, October 2013.

[10] E.M. Kornfield, R. Manmatha and J. Allan, “Text Alignment with Handwritten Documents”, 1st International Workshop on Document Image Analysis for Libraries, pp. 195-211, Palo Alto, USA, 2004.

[11] L. Lorigo and V. Govindaraju, “Transcript Mapping for Handwritten Arabic Documents”, 14th SPIE Conference on Document Recognition and Retrieval, vol. 6500, 2007.

[12] S. Zinger, J. Nerbonne and L. Schomaker, “Text-image alignment for historical handwritten documents”, Proceedings of SPIE - The International Society for Optical Engineering, Document Recognition and Retrieval XVI, 2009, vol. 7247, pp. 1-8.

[13] U.V. Marti and H. Bunke, “Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system,” Internatinal Journal of Pattern Recognition and Artificial Intelligence, vol. 15, no. 1, pp. 65–90, 2001.

[14] A. Vinciarelli and J. Luettin, “A new normalization technique for cursive handwritten words”, Pattern Recognition Letters, vol. 22, no. 9, pp. 1043-1050, 2001.

[15] T. Stafylakis, V. Papavassiliou, V. Katsouros and G. Carayannis, “Handwritten document image segmentation into text lines and words”, Pattern Recognition, vol. 43, no. 1, pp. 369–377, 2010.