Ground Truth Creation for Handwriting Recognition in Historical Documents

Andreas Fischer
Institute of Computer Science and Applied Mathematics
Neubrückstrasse 10
3012 Bern, Switzerland
afischer@iam.unibe.ch

Emanuel Indermühle
Institute of Computer Science and Applied Mathematics
Neubrückstrasse 10
3012 Bern, Switzerland
eindermu@iam.unibe.ch

Horst Bunke
Institute of Computer Science and Applied Mathematics
Neubrückstrasse 10
3012 Bern, Switzerland
bunke@iam.unibe.ch

Gabriel Viehhauser
Institut für Germanistik
Längassstrasse 49
CH-3012 Bern

Michael Stolz
Institut für Germanistik
Längassstrasse 49
CH-3012 Bern

viehhauser@germ.unibe.ch/michael.stolz@germ.unibe.ch

ABSTRACT

Handwriting recognition in historical documents is vital for the creation of digital libraries. The creation of readily available ground truth data plays a central role for the development of new recognition technologies. For historical documents, ground truth creation is more difficult and time-consuming when compared with modern documents. In this paper, we present a semi-automatic ground truth creation proceeding for historical documents that takes into account noisy background and transcription alignment. The proposed ground truth creation is demonstrated for the IAM Historical Handwriting Database (IAM-HistDB) that is currently under construction and will include several hundred Old German manuscripts. With a small set of algorithmic tools and few manual interactions, it is shown how laypersons can efficiently create a ground truth for handwriting recognition.

Categories and Subject Descriptors

I.5.4 [**Pattern Recognition**]: Applications—*text processing*

1. INTRODUCTION

Handwriting recognition of scanned or photographed text images is still a widely unsolved problem in computer science and an active area of research. In particular, the interest in the recognition of handwritten text in historical documents has grown strongly in recent years [1]. In the context of cultural heritage preservation, many libraries all around the world have digitized their most valuable old handwritings. Examples include religious Old Greek manuscripts from Early Christianity, such as the Codex Sinaiticus at the British Library, epic Old German manuscripts from the Middle Ages, such as the Codex Sangallensis 857 at the Abbey Library of Saint Gall, Switzerland, and political English manuscripts from Modern Ages, such as George Washington's papers at the Library of Congress. In order to search and browse these large collections of historical handwritings in digital libraries based on keywords, automatic recognition and transcription or an alignment with an existing transcription is needed [17].

For the research on automatic handwriting recognition, obtaining document images and a transcription is not sufficient. Instead, the document images have to be segmented into individual text lines or words that need to be aligned with the transcription in order to train and test a recognition system. The text line or word images and their corresponding transcription constitute the ground truth needed for handwriting recognition. Note that in contrast to printed documents, a segmentation into individual letters typically is not possible for handwritten documents because of the complex connections between the letters. General recognition systems, e.g., Hidden Markov Models (HMM) [15] and BLSTM neural networks [6], do not rely on letter segmentation prior to recognition and thus, individual letter images are not necessarily part of the ground truth.

In case of modern handwritten documents, ground truth data is readily available for a number of datasets. For *online* data that is acquired with special writing devices to capture time information of the writing process, the first datasets, e.g., the UNIPEN [7] and IRONOFF[25] datasets, included labeled isolated digits, letters, and words in Western scripts. Later on, the datasets focused on whole sentences for general handwriting recognition, e.g., the IAM-OnDB [12] dataset that contains a large number of labeled English sentences from the Lancaster-Oslo/Bergen (LOB) corpus. Also, datasets for non-Western scripts became available, e.g., Japanese [18] and Indian [3]. For the more difficult task of *off-line* recognition that is solely based on handwriting images, one of the first available datasets was CEDAR [8] containing labeled images of isolated, address related words and numbers, e.g., city names, state names, and ZIP codes

in Western scripts. In fact, commercial handwriting recognition systems with a high recognition accuracy are still only available for restricted domains, such as postal address or bank check reading [23, 9]. More recent datasets were focused on whole sentences, e.g., the IAMDB [16] with English sentences from the LOB corpus. Also, non-Western scripts became available, e.g., Arabic [20].

The availability of the datasets mentioned above had a huge impact on the development of recognition technologies for modern documents and allowed a comparison of different systems on established ground truth data. In case of historical documents, however, only few datasets have become readily available yet. The reason for this is twofold. Firstly, the application domain is rather new, and secondly, the ground truth creation is more difficult and time-consuming than for modern documents. In case of modern documents, artificial data sets for handwriting recognition can be rapidly created using special forms that allow a perfect automatic text line segmentation and transcription alignment. Both tasks become more difficult for real-world historical documents. Segmentation is rendered more difficult due to the noisy background caused by the ravages of time, resulting in damaged paper or parchment and ink bleed-through. Available transcriptions often are not aligned with the individual text lines or pages of the document images. If no transcription is available at all, another problem is given by the fact that only experts can perform the time-consuming transcription of the special old languages, while for modern documents, laypersons are able to perform this task.

In the literature, one of the few readily available datasets for historical documents is presented in [22]. It contains labeled bounding boxes of individual words from 20 pages of George Washington's letters [1]. Also, cleaned and labeled word images are given that were used mainly for word spotting [22, 24]. While the availability of ground truth data is still very limited, progress can be reported for methods needed for ground truth creation. For a survey of text line extraction methods for historical documents, we refer to [11]. The issue of text line segmentation into single words has been discussed, e.g., in [14]. Also, systems for the complete recognition of historical documents can be used for ground truth creation, e.g., for an initial transcription alignment that is then subject to manual correction or for the segmentation of text lines into words. First results for automatic transcription are reported in [19] for Old Greek manuscripts using a segmentation-free approach based on character cavities that is, however, specific to the Old Greek handwriting. In [4], single writer recognition is performed on the George Washington dataset using so-called alphabet-soups. A very general systems is proposed in [26], where an HMM-based recognition system for unconstrained handwriting recognition originally developed for modern scripts is adapted for the recognition of Old German manuscripts. Also, an automatic transcription based on BLSTM neural networks was proposed for single word recognition in Old German manuscripts [5]. In fact, part of the IAM-HistDB dataset whose ground truth creation is described in this paper is used for experimental evaluation in [26, 5].

[1] `http://ciir.cs.umass.edu/downloads/`

More ground truth data clearly is necessary for the recognition of historical documents, but costly to obtain manually. Some automatic tools mentioned above are available, but they commit errors. For the ground truth, however, errors are not tolerable, because ground truth data should provide clean learning examples for the recognition systems and serve as an error-free reference to evaluate the correctness of the systems. Therefore, ground truth sometimes is created completely manually or semi-automatically. In [2], for example, automatic character matching is corrected by human interaction for typewritten historical documents.

In this paper, we propose a proceeding that performs ground truth creation for handwriting recognition in historical documents as a sequence of several steps. Some of them are completely manual and others are fully automatic with manual corrections. The proposed proceeding aims at reducing the time needed and hence the costs to create an error-free ground truth by ensuring that laypersons without special knowledge in computer science or linguistics are able to perform all required manual interactions.

The proposed proceeding has been successfully used for the creation of the IAM Historical Handwriting Database (IAM-HistDB) [2]. This dataset will contain the ground truth of about 600 Old German manuscript pages once it is finished and ready to be accessed by the community. The presented proceeding takes into account special requirements given by historical documents, i.e., noisy background processing and transcription alignment. It focuses on four properties. First, it aims at reducing manual interactions as far as possible. In particular, laypersons should be able to perform all manual interactions. Secondly, no compression is used for the original document images. This leads to a dataset of considerable size, but ensures that no image detail is lost. Thirdly, only open formats are chosen for ground truth files to be independent of specific software. Fourthly, a small set of algorithmic tools is chosen for automatic processing. This allows a fast implementation and easy customization to other databases for anyone interested in using the proposed tools.

The paper is structured as follows. In Section 2 the IAM-HistDB dataset is described, which serves as the case study for the proposed procedure. Section 3 describes the ground truth creation step by step in detail, Section 4 summarizes the resulting ground truth and comments on storage and time expense, and finally, Section 5 draws some conclusions.

2. IAM-HISTDB

In this paper, the proceeding for ground truth creation is demonstrated for the IAM Historical Handwriting Database (IAM-HistDB) that contains medieval manuscripts of the epic poem *Parzival* by Wolfram von Eschenbach, one of the most significant epics of the European Middle Ages.

There exist multiple manuscripts of the poem, written with ink on parchment or paper, that differ in writing style and dialect of the language. We consider three manuscripts listed

[2] The document images of the IAM-HistDB have been made available to the authors with restrictions on their further use, but the authors aim at making the database publicly available. Researchers interested in more details are invited to contact the authors.

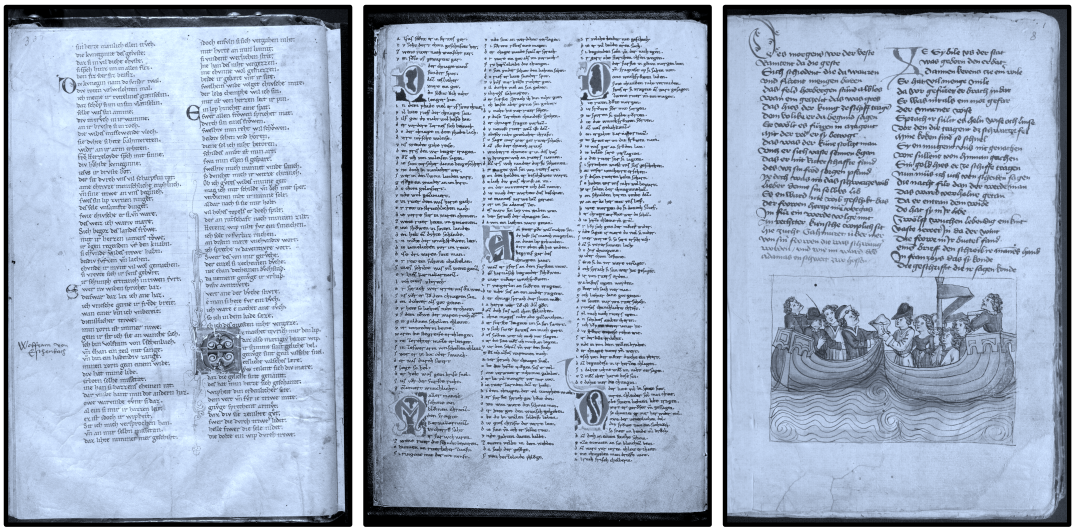

Figure 1: Examples pages from the IAM-HistDB dataset. Manuscript D, page 36, manuscript G, folium 11 (verso), and manuscript R, folium 8 (recto).

Manuscript	Codex	Century	Folia	Pages
D	Cod. 857	13th	323	646
G	Cgm 19	13th	75	150
R	Cod. AA 91	15th	180	360

Table 1: Manuscripts of the IAM-HistDB dataset.

in Table 1 that are denoted manuscript D, G, and R following the philological nomenclature. Together, the manuscripts consist of 578 sheets (folia) with a front page (recto) and a back page (verso), each. Manuscript D is kept in the Abbey Library of Saint Gall, Switzerland (Cod. 857), manuscript G in the Bavarian State Library, Germany (Cgm 19), and manuscript R in the Burgerbibliothek Bern, Switzerland (Cod. AA 91). Note that in manuscript D and G, not only the *Parzival* poem can be found. In addition, manuscript D contains, e.g., the famous *Nibelungenlied*. In Figure 1, exemplary pages are shown from each of the manuscripts that were digitized with 300 dpi. The dimension of the manuscripts is about 31 × 21 cm, each. The text is arranged in two or three columns of pairwise rhyming lines that correspond in most cases with the verses of the poem.

The IAM-HistDB dataset consists of about 600 manuscript pages for which a transcription is available. The transcriptions were acquired by the Department of German Language and Literature of the University of Bern using the *TUSTEP*[3] tool for managing transcriptions for Latin and non-Latin manuscripts. For each of the manuscripts, a digital, browser-based edition was created on DVD with a richly annotated HTML transcription, which is the basis for the ground truth transcription of the IAM-HistDB dataset.

On the *Parzival* project website of the Department of German Language and Literature [4], more detailed information about the manuscripts and their transcription can be found alongside with a number of document image previews.

[3] `http://www.zdv.uni-tuebingen.de/tustep/`

Figure 2: Text seletion using GIMP; see Figure 1, leftmost image.

3. GROUND TRUTH CREATION

In the following, the ground truth creation for the IAM-HistDB dataset is described step by step. First, text areas are identified on the document page and the text foreground is detected. Next, the text is segmented into texlines and the transcription is aligned. Finally, the textlines are segmented into words.

3.1 Text Selection

As the first step, contiguous text areas are selected with bounding polygons on each document page. In most cases, two columns are selected per page. Titles and paragraphs are included in the columns, i.e., they are not selected separately. Special cases are paragraphs covering two columns and captions of drawings. The polygon selection avoids ornaments, drawings, decorated initial letters, page numbers, and annotations on the margin of the page that were added

[4] `http://www.parzival.unibe.ch/`

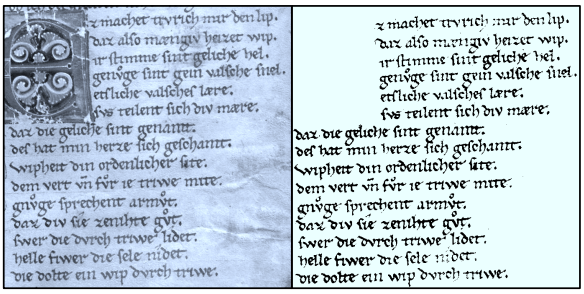

Figure 3: Binarization based on text selection and DoG enhancement.

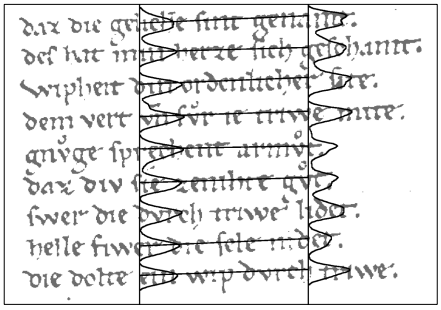

Figure 4: The skew for text line segmentation is given by the slope of the connections between the maxima of two histogramms.

later to the manuscript. For the purpose of layout analysis, these elements could be selected in a similar way and added to the ground truth in the future.

For polygon selection, the *Paths* tool of the GIMP [5] software is used and the selections are saved as Scalable Vector Graphics (SVG). While the selection is not very tight around the text areas for the purpose of fast processing, it avoids stains, holes and other artifacts on the parchment that would impede the subsequent binarization. In Figure 2, the text selection using GIMP is illustrated.

3.2 Binarization

For text line segmentation, the text foreground has to be extracted from the selected text areas. First, grayscale document images are obtained by luminance. Then, a Difference of Gaussians (DoG) filter is applied to the grayscale images to enhance the text foreground. Hereby, the image is first blurred with two Gaussian kernels having different standard deviation radii σ_1 and σ_2. Then, one of the blurred images is subtracted from the other. If the two radii σ_1 and σ_2 are chosen carefully, most of the parchment background noise, e.g., stains and ink bleed-through, can be removed and the text foreground can be accentuated. The binarization of the document image is then achieved with a global threshold T and the selected text areas are cut out from the binarized image.

For DoG filtering and thresholding, the GIMP software is used with *Script-Fu* automation. For each manuscript of the IAM-HistDB dataset, the radii σ_1 and σ_2 as well as the binarization threshold T are determined manually and are used for all manuscript pages. Hereby, a good tradeoff has to be found between reducing pepper noise and maintaining text detail. Remaining background noise typically includes holes and stitches within the text areas. Figure 3 shows a binarization example for $\sigma_1 = 40.0$, $\sigma_2 = 2.0$, and $T = 215$.

3.3 Text Line Segmentation

Once binary images of the text areas are obtained, the text is segmented automatically into text lines based on an approach that was originally proposed for online handwritten documents in [13]. The resulting piecewise linear boundary between two adjecent text lines is then subject to manual correction in a graphical user interface.

[5] `http://www.gimp.org/`

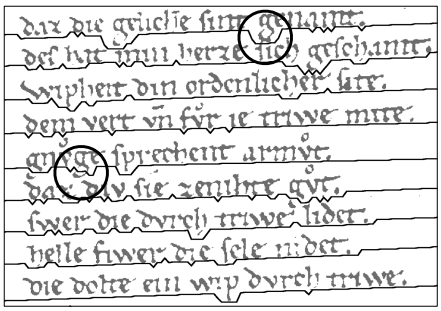

Figure 5: Automatic line segmentation. Two circles indicate segmentation errors caused by the large descender of the letter 'g'.

3.3.1 Segmentation Algorithm

The text line segmentation is performed in two steps. In the first step, the starting point and the skew, i.e., the horizontal inclination, of each text line is calculated based on histogram analysis. For the starting points, a histogram resulting from the horizontal projection of the black pixels is calculated for the left part of the text, taking only a few initial letters of each text line into account. The starting points of the text lines are then given by the maxima of the histogram. For skew estimation, two additional histograms are calculated in the middle of the text area as illustrated in Figure 4. For each text line, the skew is then given by the slope of the connection between two maxima.

In the second step, the starting points and the skew of the text lines are used to calculate a piecewise linear separating path by means of a search procedure. First, the start of the separating path is chosen in the middle between two text line starting points. Then, new points are added to the path in the direction of the text line skew at regular distance intervals. For avoiding to hit any of the two adjacent text lines, dynamic programming is used to optimize the position of the new points. Hereby, the cost function of the path position is influenced by the deviation from the direction of the skew, the distance to the foreground pixels, and a penalty for crossing a text line. In Figure 5, an example of the resulting segmentation is shown and typical errors are indicated. For more details on the text line segmentation, we refer to [13].

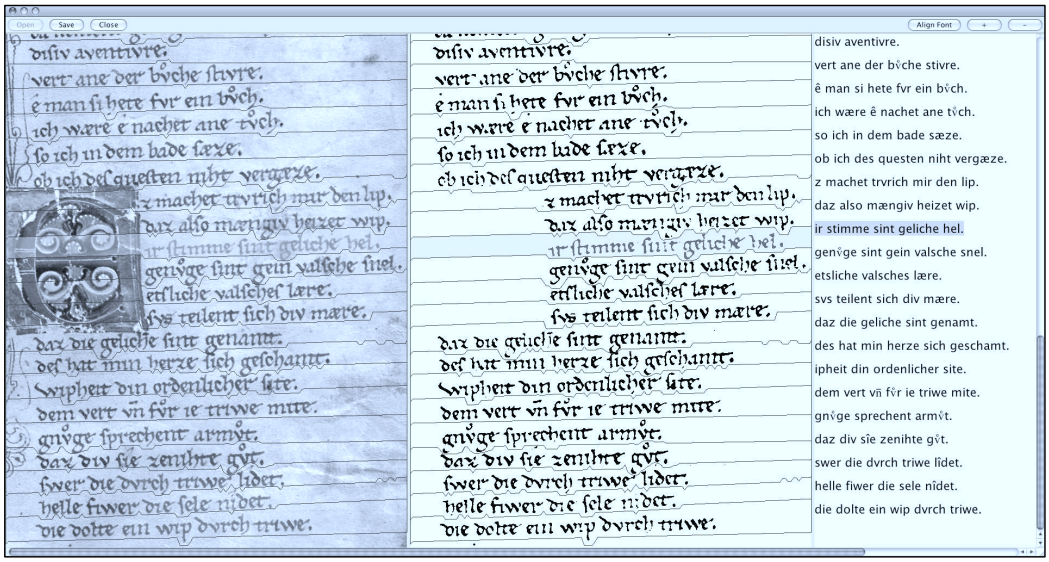

Figure 6: GUI for correcting line segmentation and transcription alignment.

3.3.2 *Manual Correction*

The text line segmentation is implemented by a Java application, which enables manual correction of the proposed segmentation path in a graphical user interface. In Figure 6, two images of a selected text column are displayed in the left and middle column, respectively. The first image is a JPG compressed partial image from the original document image and the second one is the binarization result. Both images have the same dimensions, given by the bounding box around the polygon chosen in the text selection stage. The text line segmentation result is overlaid on the images and can be corrected directly for the binarized image or, alternatively, for the color image.

After correction, the individual text lines are cut out from the binarized image for further processing. Typical problems that need manual correction include touching lines, letters with large descenders or ascenders, and special marks above a letter as illustrated in Figure 5. In the graphical user interface, two consecutive text lines share a common polygon boundary for the purpose of fast processing. For highly overlapping text lines, two different boundaries might be necessary.

3.4 Transcription Alignment

To train a handwriting recognition system, the available transcription needs to be exactly aligned with the segmented text line images. Hereby, line breaks have to be adjusted and the transcription must be stored in plain text, such that each letter in the image corresponds to one single character in the transcription. In case of multiple interpretations of characters or words, the most probable interpretation according to human experts is chosen.

First, the HTML transcription of the IAM-HistDB manuscripts is parsed and stored in Unicode plain text. Special medieval letters that are not part of the Latin alphabet are already given by a Unicode character in the HTML transcription. All elements that are not included in the selected text area, e.g., decorated initial letters and annotations on the margin of the page, are removed automatically as well as abbreviations that are written out in the transcription but appear as a special character in the text image. All information necessary for this automatic removal is given in the richly annotated HTML transcription. Line breaks are initially inserted after each verse of the poem. They correspond in most cases with the line breaks in the document image.

The parsed plain text transcription is then integrated into the Java GUI application for manual correction of the alignment as illustrated in Figure 6. In the right column, the transcription is displayed in a text editor. The text lines are distributed equally over the height of the text image by adapting font size and line spacing for visual alignment. Also, the text line transcription is synchronized with the text line images to facilitate line break correction. Note that both, the correction of text line segmentation and transcription alignment is performed at the same time using the depicted GUI application.

3.5 Word Segmentation

The last step for ground truth creation for the IAM-HistDB dataset is the segmentation of text lines into words. Following the approach presented in [27], an HMM recognition system is used to automatically determine word boundaries that are then corrected manually if necessary. Hereby, an HMM recognizer similar to the one presented in [15] is used.

3.5.1 *Text Line Normalization*

The segmented text lines are normalized prior to recognition in order to cope with different writing styles. First, the skew angle is determined by a regression analysis of the bottom-most black pixels of each pixel column. Then, the skew of the text line is removed by rotation. Finally, a vertical scaling is applied to obtain three writing zones of the

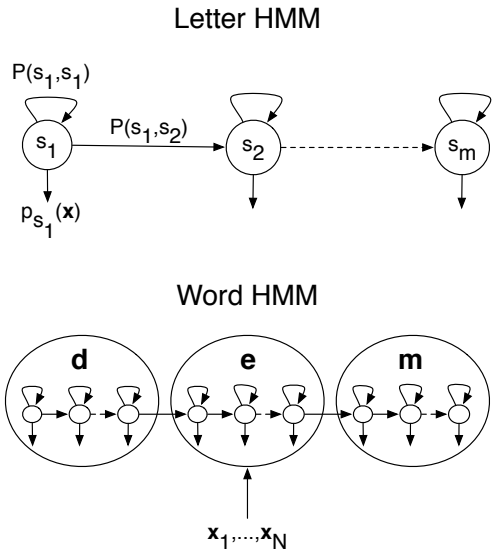

Figure 7: Hidden Markov Models.

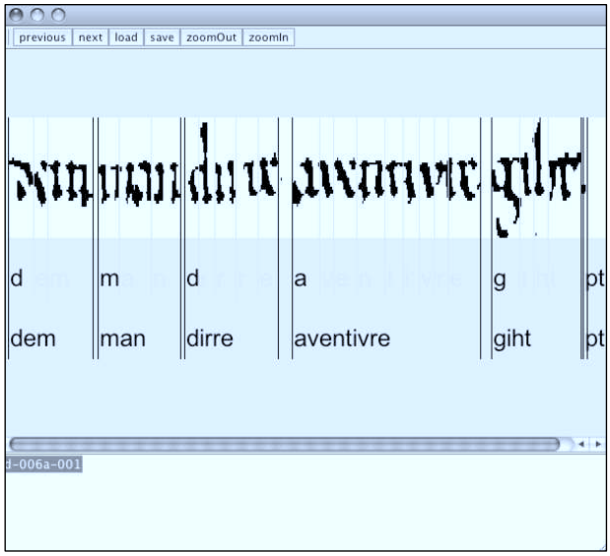

Figure 8: GUI for word segmentation correction.

same height, i.e., lower, middle, and upper zone separated by the lower and upper baseline. To determine the lower baseline, the regression result from the skew correction is used, and the upper baseline is found by vertical histogram analysis. No correction of the slant, i.e., the inclination of the letters, is performed, because almost no slant is present in the medieval handwritings. For more details on the text line normalization operations, we refer to [15].

3.5.2 *Feature Extraction*

For algorithmic processing, a normalized text line image is represented by a sequence of N feature vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N$ with $\mathbf{x}_i \in \mathbb{R}^n$. This sequence is extracted by a sliding window moving from the left to the right over the image. At each of the N positions of the sliding window, n features are extracted. As proposed in [15], $n = 9$ geometrical features are used. The sliding window has a width of one pixel. It is moved in steps of one pixel, i.e., N equals the width of the text line. From each window, three global features are extracted that capture the fraction of black pixels, the center of gravity, and the second order moment. The remaining six local features consist of the position of the upper and lower contour, the gradient of the upper and lower contour, the number of black-white transitions, and the fraction of black pixels between the contours.

3.5.3 *HMM-Based Forced Alignment*

Based on the features extracted from the normalized text line images, a Hidden Markov Model (HMM) recognizer is trained and used to find optimal word boundaries for a given transcription. This task is also known as *forced alignment*.

The HMM recognizer is based on letter models with a certain number m of hidden states $s_1, \ldots, s_m$ arranged in a linear topology that are connected to words. An illustration of a single letter HMM is given in Figure 7 (top). The states s_j with $1 \leq j \leq m$ emit observable feature vectors $\mathbf{x} \in \mathbb{R}^n$ with output probability distributions $p_{s_j}(\mathbf{x})$ given by a mixture of Gaussians. Starting from the first state s_1, the model either rests in a state or changes to the next state with transition probabilities $P(s_j, s_j)$ and $P(s_j, s_{j+1})$, respectively.

During training of the recognizer, word HMMs are built by concatenating single letter HMMs as illustrated in Figure 7 (bottom) for the word "dem". From the individual word HMMs, whole text lines can be modeled using a special model for the space character. The probability of a text line HMM to emit the observed feature vector sequence $\mathbf{x}_1, \ldots, \mathbf{x}_N$ is then maximized by iteratively adapting the initial output probability distributions $p_{s_j}(\mathbf{x})$ and the transition probabilities $P(s_j, s_j)$ and $P(s_j, s_{j+1})$ with the Baum-Welch algorithm [21]. For ground truth creation, the whole dataset is used for HMM training.

For forced alignment, the trained letter HMMs are then connected again to text lines according to the transcription. In contrast to training, the transition and output probabilities remain fixed and an optimal HMM state sequence is found with the Viterbi algorithm [21]. The corresponding feature vectors, given by the alignment, are the key to the location of the word boundaries in the text image.

For HMM training and forced alignment, the HTK[6] software is used. Important parameters of the HMM recognizer are the number of states m for each letter and the number G of Gaussian mixtures of the output probability distributions. For ground truth creation of the IAM-HistDB dataset, $m = 16$ states were used for each letter HMM and only $G = 1$ Gaussian mixture for the output probability distribution. It has been demonstrated in [10] that the best alignment can be in fact achieved with only one Gaussian mixture component. It is assumed that the importance of whitespace, which can best be modeled by one Gaussian mixture, is responsible for this result.

[6] `http://htk.eng.cam.ac.uk/`

Step	Task	Manual
1	Text selection	yes
2	Binarization	no
3	Line segmentation	no
	Segmentation correction	yes
4	Transcription alignment	no
	Alignment correction	yes
5	Line normalization	no
	Feature extraction	no
	Word segmentation	no
	Segmentation correction	yes

Table 2: Ground truth creation proceeding for handwriting recognition in historical documents.

Ground Truth Item	Description	Format
Document Images	300 dpi, colored	TIFF
Transcription	Plain text, Unicode	TXT
Text areas	Bounding polygon	SVG
Text line areas	Bounding polygon	SVG
Text line images	Normalized, binary	TIFF
Word images	Normalized, binary	TIFF

Table 3: Resulting ground truth.

3.5.4 *Manual Correction*

The results from the HMM-based forced alignment are displayed in a Java application, which enables manual correction in a graphical user interface. Only a few corrections are necessary, because the accuracy of the forced alignment typically is near 100%. In Figure 8, the Java application is illustrated for an example text line. The user can move the word boundaries to correct the alignment.

4. RESULTS

The proposed semi-automatic ground truth creation proceeding for handwriting recognition in historical documents is summarized step by step in Table 2. For details on the individual steps, see Section 3. Besides the task descriptions for each step, it is indicated whether manual interaction is involved or not. Note that all manual interactions needed for ground truth creation can be performed by laypersons.

Table 3 summarizes details of the resulting ground truth. The ground truth can be used for various document analysis tasks. The transcription together with the normalized, binary text line or the word images can be used, e.g., for testing handwriting recognition or word spotting systems. The bounding polygons of the text areas and text lines can be used, e.g., for testing layout analysis and text extraction algorithms. Also, the bounding polygons of the text lines allow a user to work directly with the original text line images, e.g., for changing binarization or normalization.

4.1 Storage and Time Expense

The main storage expense is given by the original document images that are not compressed to retain all image detail. For the IAM-HistDB dataset, the document image size is 17 MB per page for manuscript D, 88 MB per double page for manuscript R, and 18 MB per page for manuscript G. The other elements of the ground truth, i.e., transcriptions, text areas, text line areas, text line images, and word images, typically have a size of 12-64 KB per page, text line, or word.

The main time expense for ground truth creation lies in the manual selection of text areas, the correction of text line segmentation, and the correction of the transcription alignment. For the IAM-HistDB data set, we have conducted a small user study to get an impression of this time expense. Three experienced users were asked to record the time they need to process ten text columns. All three manuscripts were considered to take different column sizes into account (see Figure 1). The text area selection took about $T_1 = 2$ minutes per column on average, and the correction of the segmentation together with the correction of the alignment about $T_2 = 3.5$ minutes per column. Because the automatic alignment of the text based on the verses of the poem was near perfect, only little transcription alignment correction was needed. For the *Nibelungenlied* in manuscript D, however, this was not the case. Here, most text line alignments needed a correction of the line break and thus, T_2 was increased to about 12 minutes.

5. CONCLUSIONS

In this paper, a semi-automatic ground truth creation procedure for handwriting recognition in historical documents is presented. It takes into account noisy background that is typical for historical documents and produces a ground truth that can be used for the development and assessment of various recognition methods, e.g., automatic transcription, word spotting, or text detection. The ground truth creation is demonstrated for the IAM-HistDB dataset consisting of several hundred Old German manuscript pages.

A small set of algorithmic tools is used for the automatic part of the ground truth creation process that includes DoG-based binarization, text line segmentation using histogram analysis and dynamic programming, parser-based transcription alignment, and HMM-based word segmentation.

Manual interactions involve the selection of text areas on the original image, the correction of line and word segmentation, and the correction of the transcription alignment. Based on simple graphical interfaces, laypersons are able to perform these manual interactions efficiently. The average time needed for manual interactions for the ground truth creation of the IAM-HistDB dataset was about 5.5 minutes per text column.

The resulting ground truth consists of the original, uncompressed document images (TIFF), an aligned transcription in plain text (Unicode), bounding polygons of the text areas and the text lines (SVG), and normalized, binary text line and word images (TIFF).

Future work includes the integration of additional algorithmic tools and graphical user interfaces into a single software package and the completion and clearing of the IAM-HistDB dataset to share it with the community.

6. ACKNOWLEDGMENTS

This work has been supported by the Swiss National Science Foundation (Project CRSI22_125220/1) and by the Swiss National Center of Competence in Research (NCCR) on Interactive Multimodal Information Management (IM2). Furthermore, we would like to thank Amrei Schroettke, Chatrina Casutt, and Matthias Zaugg for creating the ground truth of the IAM-HistDB dataset.

7. REFERENCES

[1] A. Antonacopoulos and A. Downton (eds.). Special issue on the analysis of historical documents. *Int. Journal on Document Analysis and Recognition*, 9(2–4):75–77, 2007.

[2] G. Bal, G. Agam, G. Frieder, and O. Frieder. Interactive degraded document enhancement and ground truth generation. In B. Yanikoglu and K. Berkner, editors, *Document Recognition and Retrieval XV*, volume 6815 of *Proc. SPIE*, 2008.

[3] U. Bhattacharya and B. Chaudhuri. Handwritten numeral databases of Indian scripts and multistage recognition of mixed numerals. *IEEE Trans. PAMI*, 31(3):444–457, 2009.

[4] S. Feng, N. Howe, and R. Manmatha. A hidden Markov model for alphabet-soup word recognition. In *Proc. IEEE Int. Conf. on Frontiers in Handwriting Recognition (ICFHR 2008)*, pages 210–215, 2008.

[5] A. Fischer, M. Wüthrich, M. Liwicki, V. Frinken, H. Bunke, G. Viehhauser, and M. Stolz. Automatic transcription of handwritten medieval documents. In *Proc. 15th Int. Conf. on Virtual Systems and Multimedia*, volume 1, pages 137–142. IEEE, September 2009.

[6] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber. A novel connectionist system for improved unconstrained handwriting recognition. *IEEE Trans. PAMI*, 31(5):855–868, 2009.

[7] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, and S. Janet. UNIPEN project of on-line data exchange and recognizer benchmarks. In *Proc. 12th Int. Conf. on Pattern Recognition (ICPR)*, volume 2, pages 29–33, 1994.

[8] J. J. Hull. A database for handwritten text recognition research. *IEEE Trans. PAMI*, 16(5):550–554, 1994.

[9] S. Impedovo, P. Wang, and H. Bunke, editors. *Automatic Bankcheck Processing*. World Scientific, 1997.

[10] E. Indermühle, M. Liwicki, and H. Bunke. Combining alignment results for historical handwritten document analysis. In *10th Int. Conf. on Document Analysis and Recognition*, pages 1186–1190, 2009.

[11] L. Likforman-Sulem, A. Zahour, and B. Taconet. Text line segmentation of historical documents: a survey. *International Journal on Document Analysis and Recognition (IJDAR)*, 9(2-4):123–138, 2007.

[12] M. Liwicki and H. Bunke. IAM-OnDB - an on-line english sentence database acquired from handwritten text on a whiteboard. In *Proc. 8th Int. Conf. on Document Analysis and Recognition (ICDAR)*, volume 2, pages 956–961, 2005.

[13] M. Liwicki, E. Indermühle, and H. Bunke. Online handwritten text line detection using dynamic programming. In *Proc. 9th Int. Conf. on Document Analysis and Recognition*, volume 1, pages 447–451, 2007.

[14] R. Manmatha and J. L. Rothfeder. A scale space approach for automatically segmenting words from historical handwritten documents. *IEEE Trans. PAMI*, 27(8):1212–1225, 2005.

[15] U.-V. Marti and H. Bunke. Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system. *Int. Journal of Pattern Recognition and Art. Intelligence*, 15:65–90, 2001.

[16] U.-V. Marti and H. Bunke. The IAM-database: an English sentence database for off-line handwriting recognition. *Int. Journal on Document Analysis and Recognition*, 5:39–46, 2002.

[17] G. Nagy and D. Lopresti. Interactive document processing and digital libraries. In *Proc. 2nd Int. Workshop on Document Image Analysis for Libraries (DIAL 2006)*, pages 2–11. IEEE Computer Society, 2006.

[18] M. Nakagawa and K. Matsumoto. Collection of on-line handwritten Japanese character pattern databases and their analysis. *Int. Journal on Document Analysis and Recognition*, 7(1):69–81, 2004.

[19] K. Ntzios, B. Gatos, I. Pratikakis, T. Konidaris, and S. J. Perantonis. An old Greek handwritten ocr system based on an efficient segmentation-free approach. *International Journal on Document Analysis and Recognition (IJDAR)*, 9(2):179–192, 2007.

[20] M. Pechwitz, S. Maddouri, V. Maergner, N. Ellouze, and H. Amiri. IFN/ENIT - database of handwritten Arabic words. In *Proc. on CIFED*, pages 129–136, 2002.

[21] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, Feb. 1989.

[22] T. M. Rath and R. Manmatha. Word spotting for historical documents. *Int. Journal on Document Analysis and Recognition*, 9:139–152, 2007.

[23] S. Srihari, Y. Shin, and V. Ramanaprasad. A system to read names and addresses on tax forms. *Proc. IEEE*, 84(7):1038–1049, 1996.

[24] K. Terasawa and Y. Tanaka. Slit style HOG features for document image word spotting. In *10th Int. Conf. on Document Analysis and Recognition*, volume 1, pages 116–120, 2009.

[25] C. Viard-Gaudin, P. M. Lallican, P. Binter, and S. Knerr. The IRESTE on/off (IRONOFF) dual handwriting database. In *Proc. 5th Int. Conf. on Document Analysis and Recognition (ICDAR)*, pages 455–458, 1999.

[26] M. Wüthrich, M. Liwicki, A. Fischer, E. Indermühle, H. Bunke, G. Viehhauser, and M. Stolz. Language model integration for the recognition of handwritten medieval documents. In *Proc. 10th Int. Conf. on Document Analysis and Recognition*, volume 1, pages 211–215. IEEE, July 2009.

[27] M. Zimmermann and H. Bunke. Automatic segmentation of the IAM off-line database for handwritten English text. In *Proc. 16th Int. Conf. on Pattern Recognition*, volume 4, pages 35–39, 2002.