International Journal on Document Analysis and Recognition manuscript No.
(will be inserted by the editor)

Video-based Whiteboard Reading

Markus Wienecke, Gernot A. Fink, Gerhard Sagerer

Bielefeld University, Faculty of Technology, 33594 Bielefeld, Germany

Received: date / Revised version: date

Abstract. Whiteboards are very popular tools in meeting rooms. With the increasing computational support for collaborative work-environments electronically enhanced whiteboards have been developed to serve as automatic meeting assistants. The most flexible of these systems use cameras to observe the whiteboard, and, therefore, do not require the use of special pens or erasers. However, currently these systems are only able to interpret some special graphical symbols and can not produce a transript of the documents written on them. As a major advancement beyond the state-of-the-art we propose a system for automatic video-based reading of unconstrained handwritten text from a whiteboard. Text lines are extracted from the captured image sequence using an incremental processing strategy. The recognition results are then obtained from the text-line images by off-line techniques and a "segmentation-free" statistical recognizer. We will present results of writer independent experiments for both lexicon-based and lexicon-free recognition of unconstrained handwriting, which demonstrate the effectiveness of our approach.

1 Introduction

In the field of human-computer interaction the ultimate goal is to make interaction with computing machinery possible without the need for special input devices such as keyboard or mouse. Therefore, more natural input modalities of communication such as speech, gesture, and handwriting are extensively studied. Besides investigating the use of handwriting for interacting with PDAs there is also a growing interest in systems for recognizing handwriting on the whiteboard. This is mainly due to the increasing popularity of whiteboards not only for presentations and educational purposes but also in meeting rooms for the exchange of ideas during group discussions, for project planning, system design, etc. Therefore, systems that can serve as automatic meeting assistant to support collaborative working are highly desirable.

In order to make use of the whiteboard as a user interface for human computer interaction, systems based on electronic whiteboards have been developed. Similar to digitizing tablets these systems employ electronic pens and erasers allowing their position in the plane to be sensed and tracked during the writing process. Using a computer the movements

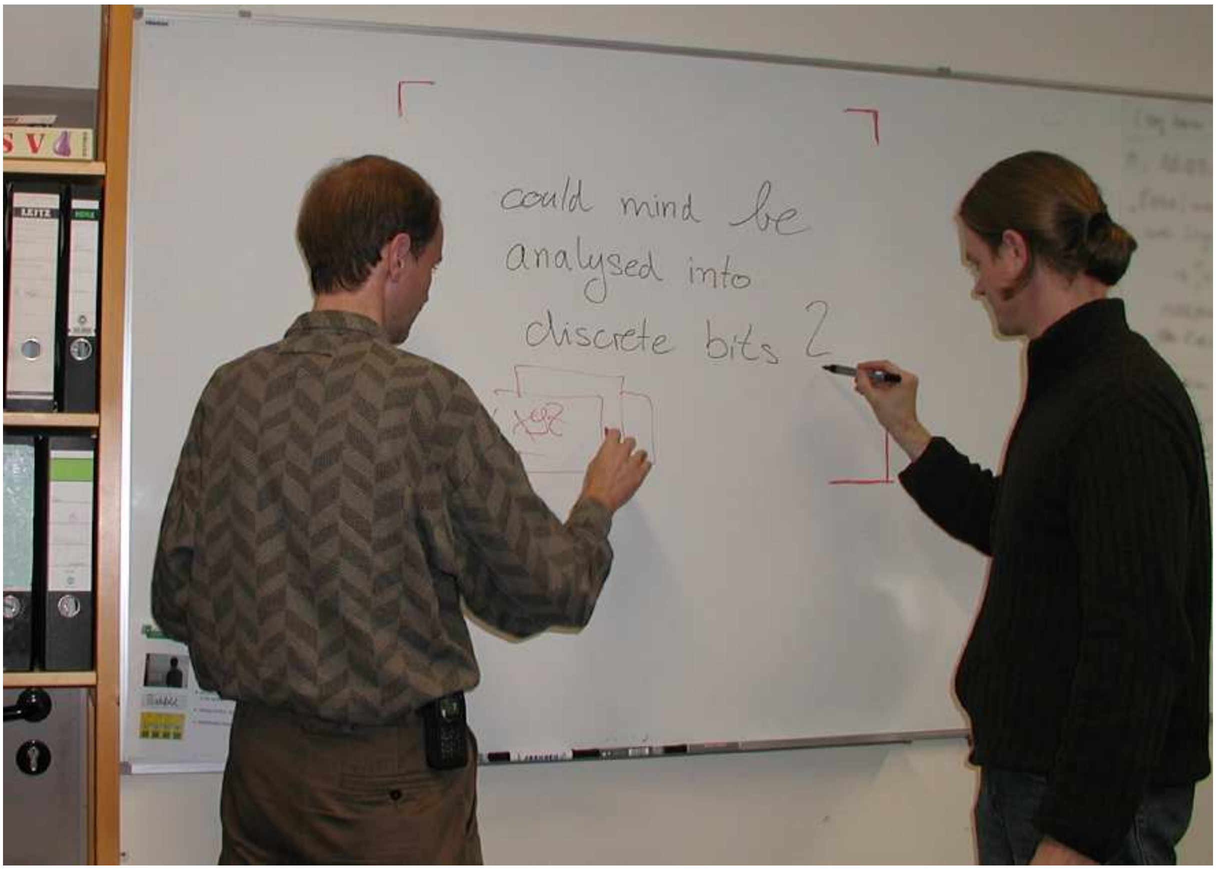


Fig. 1. Whiteboard as a user interface for collaborative working

of the pen and the eraser can be transformed in order to construct an electronic version of the document-image on the whiteboard. Additionally, the pen trajectory can be interpreted by an on-line recognition module to automatically recognize what was written on the board.

However, electronic whiteboards exhibit some disadvantages. As special pens and erasers are necessary, the natural interaction is restricted. For example, an electronic whiteboard does not notice the erasing of parts of the written document, if the writer uses a finger or paper towel instead of the eraser provided. Furthermore, graphical symbols or text written on Post-It notes which are then affixed to the board cannot be recognized. Therefore, a promising alternative might be to retain ordinary whiteboard and pens and to observe the writing process using a video camera.

In order to cover a large area of the whiteboard, the preferable position of the video camera is several meters in front of the board, either mounted to the ceiling or fixed on a tripod. With this setup, the gestures of the user writing on the board could also be considered. Furthermore, an active camera together with mosaicing techniques could be used to enlarge the observable area. Unfortunately, this setup also has a severe disadvantage. As the user usually stands in front of the board to have a clear view during the writing process, the pen and portions of text are very often occluded by the writer and therefore not visible in the image sequence. In order to circumvent this drawback, a kind of activity analysis could be employed to decide whether the captured image is suitable for further processing. An alternative method is to extract only the visible portions of the handwritten text and to incrementally integrate the partial results to the overall recognition result.

In this paper a system for automatic video-based whiteboard reading is presented. In contrast to the approaches proposed in [25, 22] which only permit the recognition of a limited set of symbols, our system is designed for recognizing unconstrained handwritten text. As the pen is rarely visible in the image and thus online recognition based on the pen trajectory is not feasible, the proposed system is characterized by an incremental off-line recognition approach. Thus, the writing process is continuously observed and recognition starts automatically as soon as a region of handwritten text is visible in the image. Besides saving processing time as only small portions of text are recognized at each time, this approach also allows obtaining a qualitative estimation of the time-structure of the handwriting process. This is an important prerequisite for systems used for meeting assistance and e-learning as the handwriting can be related to the gestures or the speech of the user.

In the following section we will give a review of relevant related work. The architecture of the proposed whiteboard reading system is presented in section 3. Afterwards we describe in subsequent sections the methods for text extraction, pre-processing, feature extraction, and statistical modeling and recognition. Evaluation results will be presented in section 9 in order to demonstrate the effectiveness of the proposed approach.

2 Related Work

In the field of human computer interaction natural input modalities like speech, handwriting, and gestures have been extensively studied in recent years. Thus, as whiteboards are popular tools in meeting rooms and the computing machinery has become increasingly powerful, there has been a growing interest in making use of the whiteboard as a user interface for human computer interaction. Therefore, systems have been developed in order to serve as meeting assistants for e.g. collaborative working. Usually, such systems are based on electronic whiteboards and special pens and erasers, which are capable of detecting the current pen position during writing in order to construct an electronic version of the image in the computer for further processing (cf. e.g. [7]). However, a severe drawback of such systems is the restricted natural interaction as special pens and erasers are required.

2.1 Video-based Systems

In order to circumvent the drawbacks that are related to specialized hardware it was proposed to retain the ordinary whiteboard and to use a video camera for observing the board. In contrast to electronic whiteboards where the pen trajectory is directly recorded, systems based on visual input first have to detect pen movements or relevant image regions in the image sequence.

One approach is to use a special marker for writing that has a distinctive color. By tracking the pen a temporal trajectory is obtained that can be recognized using on-line methods. In [1] a system based on visual input is described, which allows the user to control the computer through simple gestures. It uses a color histogram tracker to obtain the trajectory of a special marker pen. The resulting trajectory is then classified by a kind of particle filtering algorithm in order to recognize the gestures. A video-based system which is capable to track an ordinary pen in image sequences was proposed by Munich & Perona [19]. The tracking process is based on a template matching approach, i.e. the position of the pen tip is found by maximizing the correlation of the pen template and the image. Additionally, a kinematic motion model is applied to constrain the search space for template matching to a relatively small image region. In [11,27] the trajectories obtained from the template matching approach described above are used for online handwriting recognition. A similar handwriting recognition system based on visual input was proposed by Bunke et al. [2]. Here, the pen trajectory is obtained by analyzing the difference image of two consecutive frames. These online systems can be successfully employed in scenarios, where the pen is always visible in the image, as this is a necessary precondition of such systems. However, they can hardly be applied for whiteboard reading where the pen is very often occluded by the writer, who usually stands between the board and the camera.

Therefore, a contrary approach for video-based whiteboard reading is to extract and analyze the relevant image regions after the writing process has finished. For example, the video-based *BrightBoard* system described in [25] continuously observes the whiteboard and grabs a suitable image when the movement of the writer has finished. The image is then analyzed in order to find and recognize graphical marks that correspond to commands allowing the user to control the computer. A similar camera-based whiteboard scanner is the so called *ZombieBoard* system proposed in [22], which applies a mosaicing algorithm to enable high-resolution whiteboard imaging. The system also monitors activity in front of the board and watches the users to draw graphical marks indicating commands and associated parameters.

2.2 Handwritten Text Recognition

As the system presented in this paper is not restricted to a small set of commands but is designed for recognizing unconstrained handwritten text the approach is not only closely related to the task of locating text in image sequences but also to off-line handwriting recognition. Whereas the problem of text detection in image sequences was so far mostly studied for machine printed text [13,18,5], there has been a lot of work in the field of off-line handwriting recognition. See e.g. [26,21] for an extensive survey.

Systems for isolated word recognition using a small lexicon achieve high recognition rates and are therefore successfully employed for the task of postal address and legal amount reading. In contrast to these tasks the recognition of unconstrained handwritten texts using a large or even unlimited vocabulary is much more difficult. This is mainly caused by the absence of context knowledge and word segmentation information.

Despite of these difficulties, several systems for unconstrained handwritten text recognition have been developed. Earlier approaches applied segmentation-based methods in combination with sophisticated classification techniques (for a survey see e.g. [3]). In more recent work, however, segmentation-free methods were pursued in order to avoid errors introduced by segmenting the text into words or even characters at an early stage. Here, Hidden-Markov-Models (HMMs) were successfully applied and gained growing interest in the research community. In [14] a segmentation-free method based on HMMs was proposed where a whole line of text is fed into the recognition module. The system is tested in single writer mode and achieves promising recognition results by incorporating statistical language models. Advanced systems for writer-independent unconstrained text recognition which are also tested on a large database [15, 17] produced by several hundreds of writers can be found in [16, 28].

3 System Architecture

The system presented in this paper is characterized by an incremental processing strategy. The writing process is continuously observed and the recognition process is activated as soon as a handwritten text region is visible in the image. Thus, the text regions are classified in their order of appearance and integrated into the overall recognition result. An example of the recognition process is shown in figure 2.

The architecture of the proposed system is depicted in figure 3. After grabbing the image, all text regions currently visible are extracted. In order to avoid recognizing the same text region multiple times in the image sequence, we employ a region memory containing all the different text regions extracted so far. If a new, not yet memorized text region is found, several pre-processing steps are applied to compensate for the highly varying background intensity and to normalize the handwriting. After that, features are extracted using a sliding window approach which are finally fed into a statistical recognition module based on Hidden Markov Models (HMMs).

4 Image Aquisition

The image sequences required for whiteboard reading are captured using a standard video camera (Sony EVI-D31) mounted on a tripod positioned approximately 3 to 4 meters in front of the whiteboard. The observed writing space covers an area of approximately 70 × 50 cm of the board. The camera is working in interlaced PAL mode grabbing about 5 images per second at a size of 756 × 576 pixels. From this it follows, that the effective resolution is less than 30 dpi, which is about ten times smaller compared to scanned documents mostly used for handwriting recognition. The observed writing space could be enlarged using an active camera and mosaicing techniques (cf. e.g. [22, 18]), but as the construction of mosaics is a time consuming process we use a fixed camera in order to achieve short response times.

5 Text Detection

5.1 Requirements

As the extraction of the handwritten text regions in the image sequence is an essential module for further processing some requirements have to be met. First of all, the extraction of the text regions has to be robust with regard to noisy images, i.e. non-text regions that belong to the writer or are caused by non-uniform lighting should be suppressed. Additionally, the text extraction has to avoid splitting up words in order to facilitate lexicon-based recognition. Furthermore, graphical marks as lines, arrows, or circles have to be detected, as

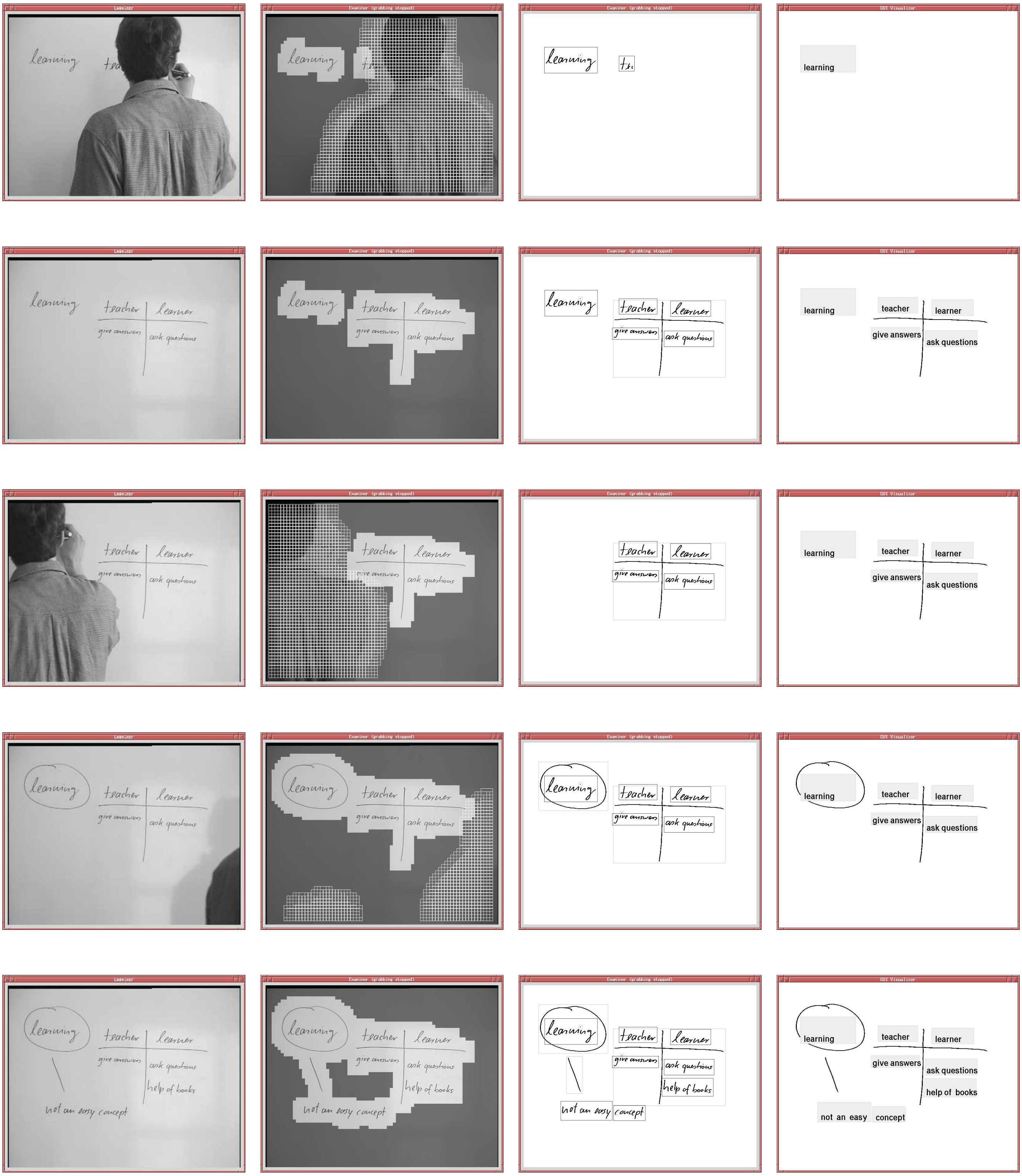


Fig. 2. Snapshots of an example image sequence. First column: grabbed images. Second column: distinction of text, background and noise regions based on the block partition of the image. Third column: clustering of binarized text components to words or phrases, respectively. Fourth column: results of text recognition.

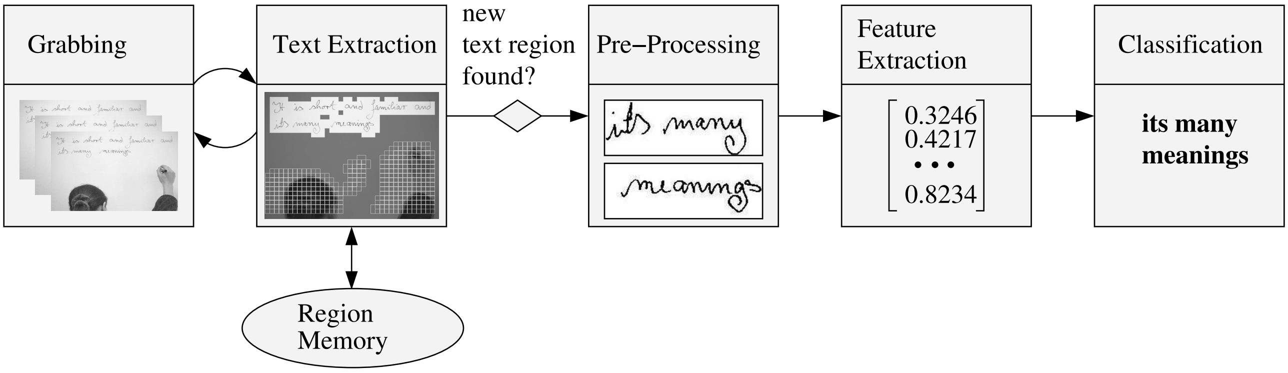


Fig. 3. System architecture

they are typically used in tables and diagrams or to emphasize words. Another important requirement is a short processing time that keeps the delays between writing and recognition as short as possible. In order to satisfy these conditions a two-step approach for robust and fast extraction of handwritten text regions is used (see figure 4(a)-4(d)).

5.2 Discrimination of Text, Background, and Noise

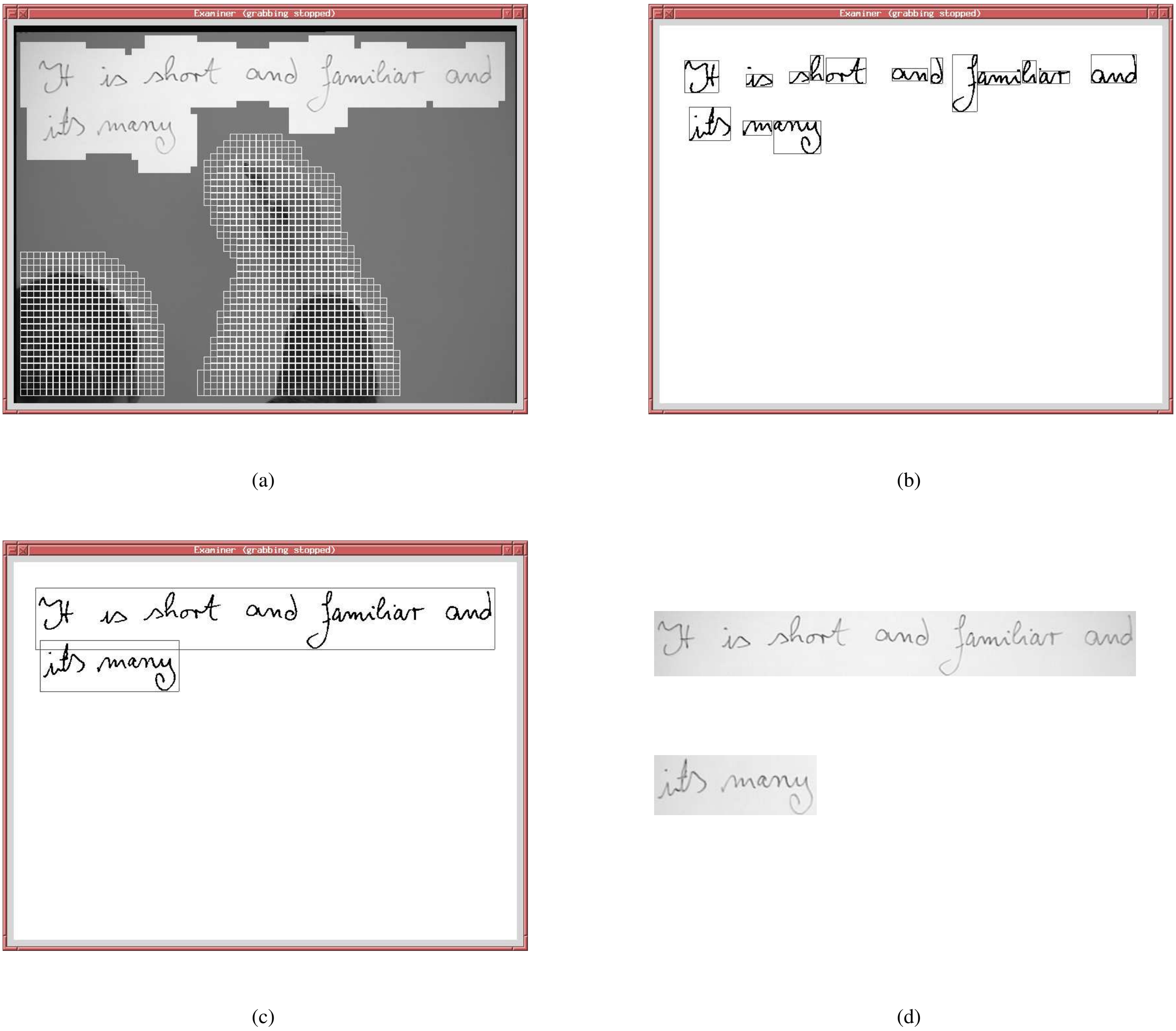
In the first step of the text extraction process the gray-scale image is divided into overlapping blocks of equal size (40 × 40 pixels). On each of these blocks a three dimensional feature vector is calculated for discriminating text from noise and pure background regions. It is assumed that text blocks can be identified using the following characteristics: They contain contour pixels caused by the pen strokes, show an average pixel intensity similar to the empty whiteboard, and, compared to noise regions, remain relatively stable over time. Therefore, the feature vector consists of the following three components:

1. the average pixel intensity of the block
2. the average difference of pixel intensity between two consecutive images per block
3. the number of edge pixels per block (determined using a Sobel edge detector)

Empirically determined thresholds are then applied to the components of those feature vectors in order to decide whether the image block is part of a text region, an ’empty’ whiteboard region, or noise.

5.3 Aggregation of Text Components

After the coarse discrimination of text, background, and noise regions was carried out, the image blocks, which are assumed to contain text and are sufficiently far away from noise regions as well, are binarized. Here, the Otsu method [20] using one global threshold per block is employed to achieve a fast binarization. Subsequently, the connected components associated to ink strokes (the black pixels) are calculated. The connected components of all text blocks are then clustered based on the distances between their bounding boxes in order to obtain regions corresponding to handwritten words or text lines. If adjacent text lines are touching, i.e. they are sharing at least one connected component, the horizontal projection profile of the image region is searched for a minimum in order to split the lines. After having determined the bounding



(a)

(b)

(c)

(d)

Fig. 4. Detection of text regions: (a) Distinction between text, background and noise regions. (b) Connected components. (c) Bounding boxes of aggregated text components. (d) Extracted text regions.

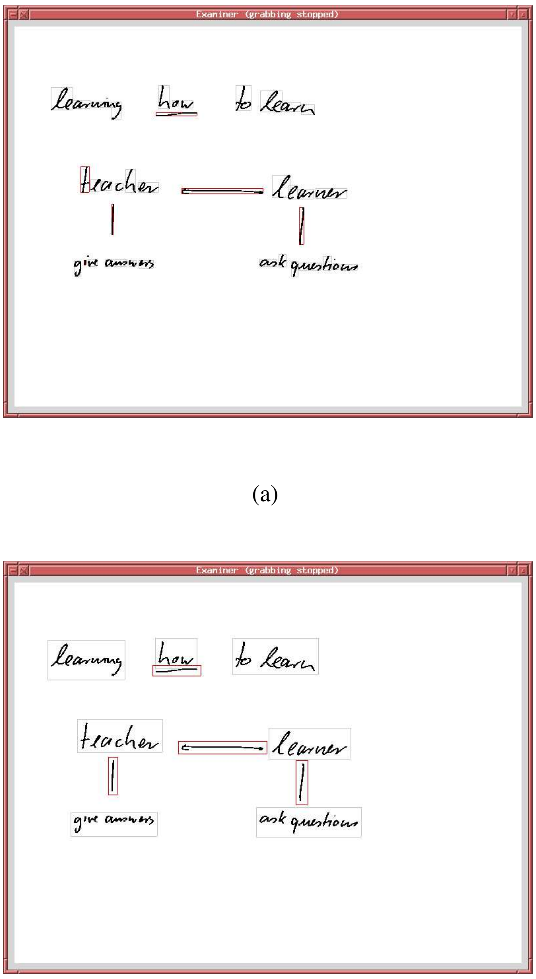
boxes of the aggregated text regions, the corresponding region is cut out of the original image for further processing.

In order to avoid recognizing the same text region repeatedly in consecutive images a region memory is employed that contains all regions recognized so far. Thus, in each time-step the extracted regions are compared with the regions stored in the memory. If the similarity (based on the pixel-wise difference of the connected component representation) exceeds a threshold it is assumed that the newly extracted region has already been recognized previously and, therefore, will not be recognized again. The extracted text regions that cannot be found in the memory are then used for further pre-processing.

Besides increasing the processing speed, the region memory permits the handling of corrections – an important feature for human-computer interaction. The handling of corrections is also accomplished by permanently comparing the region memory with the current image. If the writer wipes out a portion of text, the associated region is obviously no longer visible in the image. From this it follows, that the image region which corresponds to the region previously stored in the memory cannot be found. Therefore, the region will be deleted from memory, so that a different portion of text can be recognized instead.

5.4 Detection of Graphical Marks

For the detection of graphical marks a couple of heuristics are used based on the connected component representation of the ink strokes. We first calculate the size of the bounding box and the *lineness* of a connected component, i.e. the relation of the number of ink pixels to the length of the diagonal of the bounding box. Obviously, the lineness is small for straight lines and increases with the curvature of the ink stroke. The lineness feature in conjunction with size and the distance to adjacent components is, therefore, well suited to detect isolated straight lines as they are used in diagrams and tables or for underlining words. In figure 5(a) the connected component representation of an example input image is shown. The components which are assumed to correspond to graphical marks are emphasized. The resulting aggregated components are shown in figure 5(b). Note that the component “t” of the word teacher is correctly aggregated to the text region because of the proximity to adjacent text components.



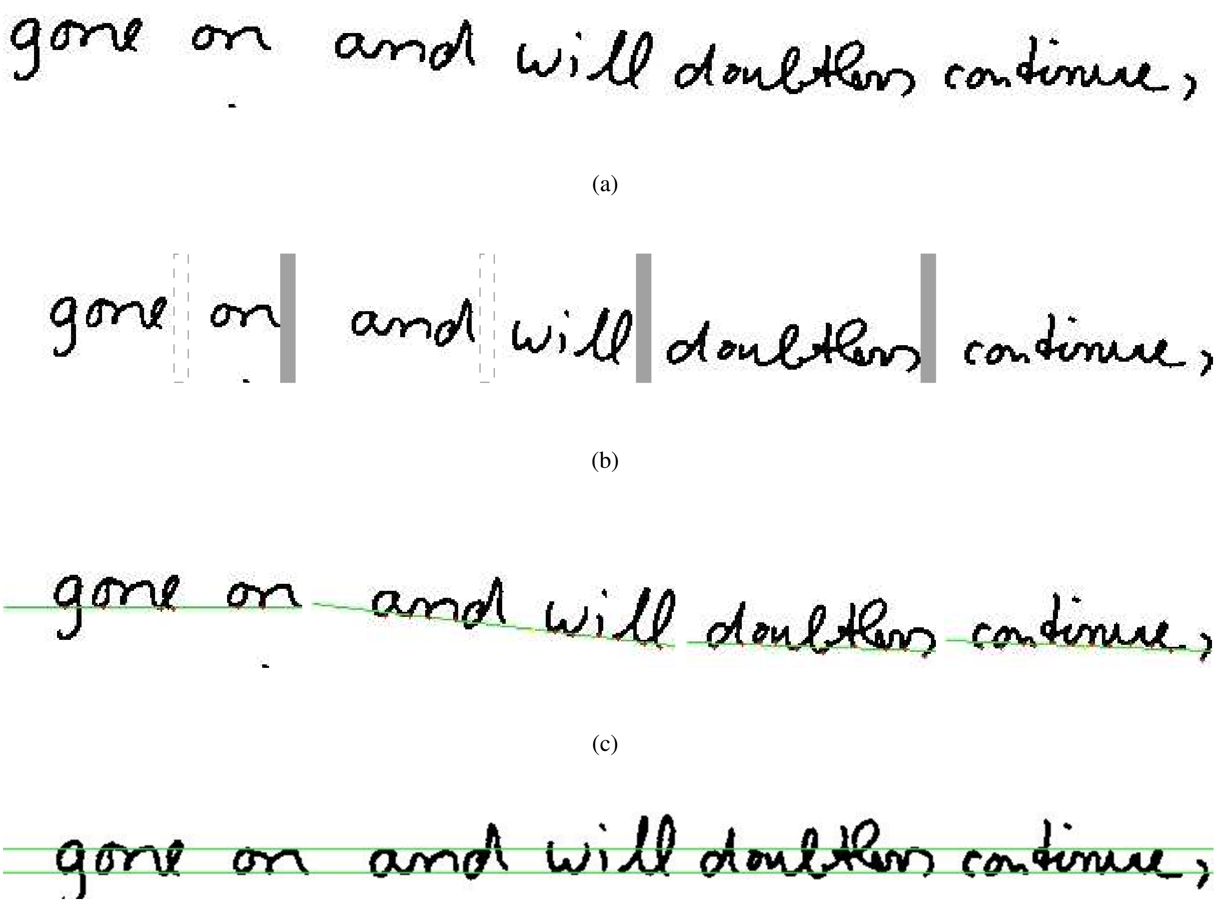
(a)

(b)

Fig. 5. Detection of graphical marks. (a) Connected components and (b) aggregated components. Components corresponding to graphical marks are emphasized.

6 Preprocessing

It is a well-known fact that the normalization of handwriting has a great impact on recognition accuracy. This particularly applies to the task of video based whiteboard reading which is much harder than reading scanned documents. One reason is that the illumination conditions often cause severe difficulties. As no specialized lighting is employed the background intensity of the image is highly varying so that no



(a)

(b)

(c)

(d)

Fig. 6. Local normalization of the handwriting. (a) The binarized text line. (b) Detection of segments. The discarded splitting hypotheses are shown in dashed lines. (c) Local baselines. (d) Resulting text line with upper and lower baseline.

global threshold exists for discriminating between the foreground and the background. Therefore, a modified version of Niblack's binarization method presented in [29] is used for determining thresholds locally. The calculation of the local threshold $t(x,y)$ is based on the average pixel intensity $\mu(x,y)$ and the standard deviation $\sigma(x,y)$ of pixel intensity in the local neighborhood:

$$t(x,y) = \mu(x,y) + k\left(\mu(x,y)\left(1 - \frac{\sigma(x,y)}{R}\right)\right).$$

The dynamic range of the pixel intensities is denoted by R. The parameter k depends on the application. We found out empirically, that a value of 0.06 for k is well-suited for our application.

Besides the problems concerning the illumination, another difficulty of whiteboard reading is the geometrical distortion of the handwriting. The lack of any reference lines together with the circumstance that subjects are often not very familiar with writing on boards results in distorted patterns of handwriting. Typically, long lines of text often show drifts of the baseline (see figure 6(a)). Motivated by this observation, the vertical position, skew, and slant of each text region are corrected locally. Therefore, each line is split by searching for white-spaces between the segments of handwriting (see figure 6(b)). A splitting position is hypothesized, if a whitespace is found that spans at least ten pixels. In order to avoid segments that are too short for calculating reliable normalization factors, we discard those splitting hypotheses which would result in segments shorter than 100 pixels.

A local baseline is then calculated for each of the segments of the handwritten line (see figure 6(c)). The following three step procedure is applied to achieve a robust baseline estimation. Firstly, the horizontal projection histogram is calculated in order to coarsely estimate the position of the body of the writing i.e. the area between the upper and lower baseline. In the second step, the local contour minima of the writing are extracted for computing a straight line approximation using linear regression. A distance threshold is applied to discard those minima which are too far away from the body of the writing. In the last step the estimated position of the baseline is improved by discarding further outlier minima. Here, a minimum is assumed to be an outlier, if its distance to the baseline is at least two times larger than the average distance. The remaining minima are then used for estimating the final position of the baseline. After having determined the local baselines, the orientation and vertical displacement of the segments are corrected by a rotational and translational transformation in order to align the local baselines to a global horizontal line (see figure 6(d)).

Subsequently, the segments of the handwritten line are used for local slant normalization. The method for calculating the local slant angle is based on the edge orientation information of the respective segment similar to the approach described in [24]. Firstly, horizontal runs of ink pixels are eliminated as they do not account for the slant of the writing. Afterwards, the Canny edge detector is applied in order to obtain the edge orientation data which is accumulated in an angle histogram. The mean of the histogram is used as slant angle which can then be employed in a shear transformation to normalize the handwriting.

In order to normalize the size of the handwriting, we count the number of local extrema of each handwritten line and put this number in relation to the width of the line. The scaling factor is based linearly on this relation, because the larger this relation the narrower the writing-style.

7 Feature Extraction

The pre-processed images are used as input data for the feature extraction step. A sliding window technique is applied similar to the approach described in [16]. In our case, a window of the image's height and four columns width is moved with an overlap of two columns from left to right over the image and several geometrical features are extracted.

As the word contour is assumed to be an important feature for reading (cf. e.g. [23]), the average distance of the lower baseline to the upper contour as well as to the lower contour are calculated (figure 7(a)-(b)). Additionally, the distance of the center of gravity of the ink pixels to the baseline is computed (figure 7(c)). These features are then normalized by the core size, i.e. the distance between upper and lower baseline, in order to increase the robustness against variations in writing-size.

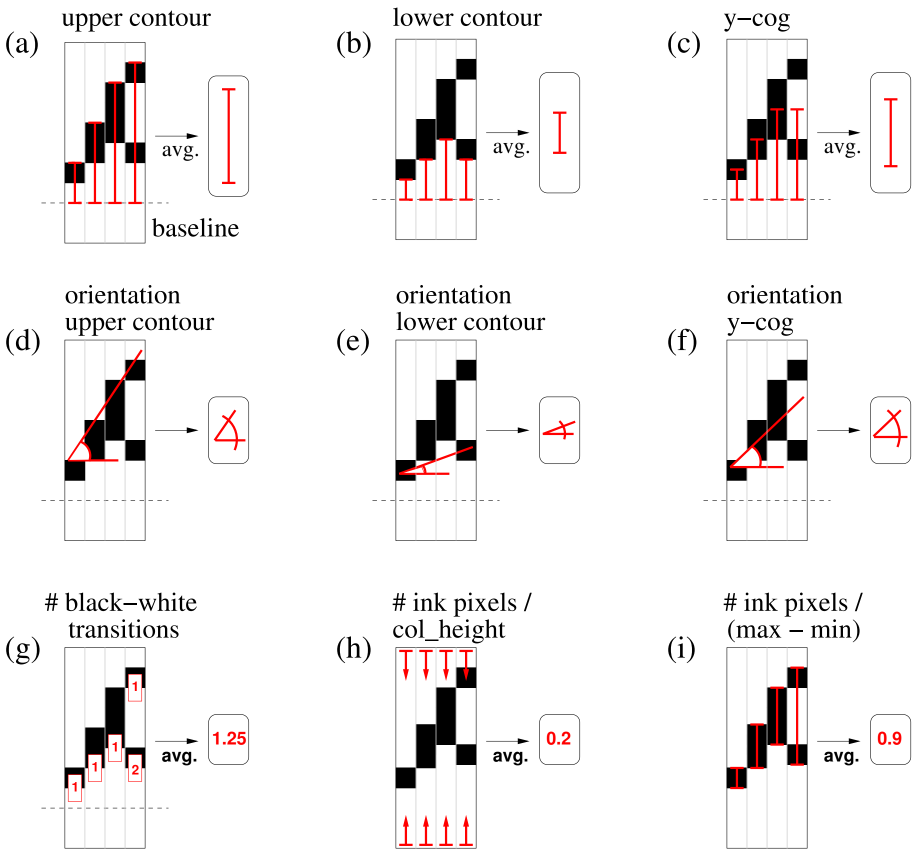


Fig. 7. Feature extraction. (a)-(c) Positional features of the contour and the center of gravity (cog). (d)-(f) Orientational features. (g) Average number of black-to-white transitions. (h)-(i) Features related to the number of ink pixels.

In order to consider the direction of the lower and upper contour as well as the gradient of the mean value of the pixel distribution, we additionally calculate three directional features. Therefore, we estimate straight lines by linear regression through the four lower contour points, upper contour points, and mean values within the sliding window. The line orientations with regard to the baseline are then used as features (figure 7(d)-(f)).

Furthermore, we calculate the average number of black-to-white transitions per column, the average number of ink pixels per column, and the average number of ink pixels between the upper and lower contour (figure 7(g)-(i)).

For considering a wider temporal context, we additionally compute an approximate horizontal derivative for each component of the feature vector, so that a 18 dimensional feature vector is obtained (9 features per window + 9 derivatives).

In order to decorrelate the feature vectors and to improve the class separability we integrate linear discriminant analysis (LDA) in the training and recognition phase (cf. [6]). The original feature representation is optimized by applying a linear transformation A, which is obtained by solving an eigenvalue problem using the within class scatter matrix S_w and the between class scatter matrix S_b of the training data. As for computing these scatter matrices each feature vector has to be labeled with an HMM state, we at first carry out an ordinary training followed by a state-based alignment of the training data. When the scatter matrices are known the LDA transformation is computed by solving the following eigenvalue problem

$$\mu_i (A^T \Psi)_i = S_w^{-1} S_b (A^T \Psi)_i \qquad (i = 1, 2, \ldots, m)$$

where μ_i and $(A^T\Psi)_i$ are the eigenvalues and eigenvectors of $S_w^{-1}S_b$. A reduction of the dimensionality can be obtained by taking into account only m eigenvectors belonging to the m largest eigenvalues. Here, the full dimensionality of the feature space is kept. After LDA transforming all feature vectors a completely new HMM training is carried out.

8 Statistical Modeling & Recognition

A successful statistical recognition system for handwriting or spoken language consists of two modeling components, one that describes the realization of individual segments, e.g. words or characters, and another that describes the restrictions on the expected segment sequences. The first component is usually realized by Hidden-Markov Models (HMMs) that model the probability density $p(\boldsymbol{x}|\boldsymbol{w})$ of observing a certain sequence of feature vectors $\boldsymbol{x}$ given a sequence of words or characters $\boldsymbol{w}$. The restriction of these sequences to plausible ones is achieved by defining a probability distribution $P(\boldsymbol{w})$ for all possible sequences, which can be realized by a Markov-chain or n-gram model. The goal of the recognition process is then to find the word or character sequence $\hat{\boldsymbol{w}}$ that maximizes the probability of the combined statistical model given the observed data $\boldsymbol{x}$ according to:

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmax}}\, p(\boldsymbol{x}|\boldsymbol{w}) P(\boldsymbol{w})^{\rho}$$

In analogy to the terminology used in spoken language processing the HMM $p(\boldsymbol{x}|\boldsymbol{w})$ could be termed the *writing model* and the n-gram model $P(\boldsymbol{w})$ is equivalent to the so-called *language model*. In order to adjust the contributions of both modeling components within the combined recognition model in practice a weighting factor ρ is used. The value of this constant, which is sometimes called the *linguistic matching factor*, has to determined empirically on cross-validation data.

8.1 Corpora

For the design of statistical recognition systems the availability of a sufficiently large database of training samples is an important prerequisite. Ideally, for a video-based system it would be desirable to obtain a large amount of image data recorded while observing a subject writing on the whiteboard. However, recording and labeling of such video data requires a substantial manual effort. Therefore, we decided to use the IAM-database of scanned documents [17] for training and cross-validation. The database provides a large amount of handwritten text documents that were produced by several hundred subjects based on prompts taken from the LOB-corpus [12]. The documents are divided into categories according to the different topics covered.

Unfortunately, the IAM-database does not contain writer ids for the handwritten samples. However, writers never provided samples for different categories. Therefore, we defined the training data to comprise categories A to D and the cross-validation data categories E & F. This partitioning corresponds to the training and test sets used in [28] and ensures all experiments to be truly writer independent.

Table 1. Corpora of handwritten & text data: word counts include punctuation and word fragments resulting from hyphenation; character counts include approximately 20% of white space.

	Source	Type	Categories	Documents	Writers	Lines	Words	Characters
Training	IAM-DB	scanned document	A – D	492	>200	4222	36582	189852
		text prompt	A – D	492	–	–	37273	–
Cross-validation	IAM-DB	scanned document	E – F	129	≈50	1081	9612	49002
Test	whiteboard	video document	F01	20	10	173	1171	6171

The test data was collected in our lab by recording image sequences of texts written on a whiteboard. In order to be able to compare the performance of the video-based system with our off-line recognizer [28], we asked ten subjects to write portions from the off-line cross-validation texts on the whiteboard, namely from category F01. No constraints with respect to the writing style were given. In contrast to the training patterns resulting from scanned forms, where rulers on a second sheet put below were used to align the baseline horizontally, the video-based data often show baseline drifts and variations of the corpus height.

A summary of the relevant characteristics of the corpora used is given in table 1. Figure 8 shows examples of a scanned document used for training and the final version of a video document from the test data. Additionally, the results of the incremental text detection are shown, which, for the example given, produces an additional segment for the third text line of the original document.

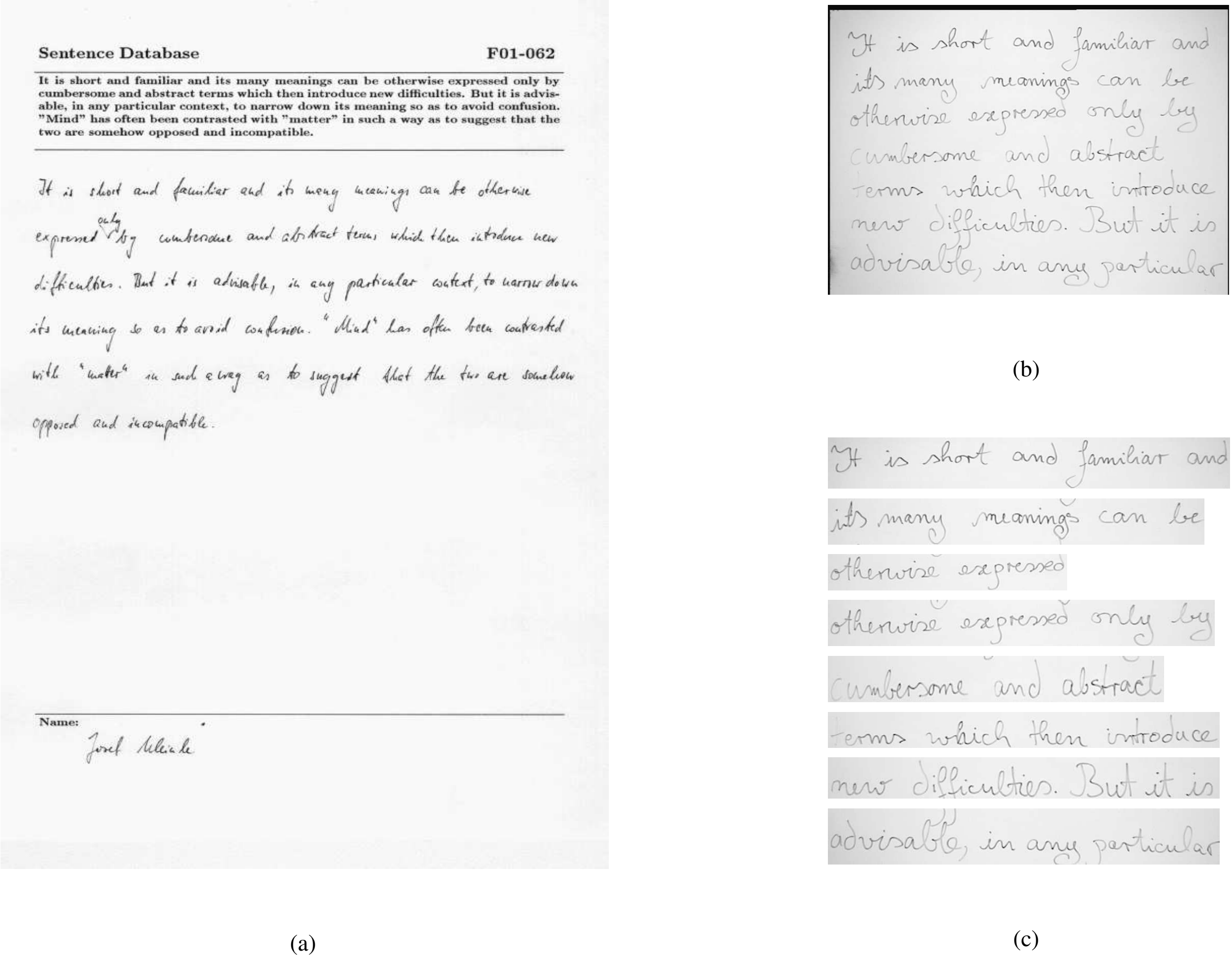
8.2 Writing Model

The configuration and parameter estimation for the HMMs defining the writing model as well as for the language models used is carried out in the framework of the ESMERALDA development environment [8].

As general setup we use semi-continuous HMMs with a shared codebook of approximately 2000 Gaussian mixtures with diagonal covariance matrices. A total of 75 HMMs are created for modeling 52 letters, ten numbers, twelve punctuation marks and brackets, and white space. The later consists of three variants accounting for different lengths in blank space between words or characters. All these models use the *Bakis*-type topology, i.e. they are basically linear models which in addition to loops and forward state-transitions permit the skipping of states in the sequence. Thus, the models can cope with a wider range of lengths in the character patterns described.

The shared codebook is initialized in unsupervised mode by applying the *k*-means algorithm to the training data. Then the initial HMM parameters can be determined on labeled initialization data. Afterwards, we apply several iterations of the Baum-Welch parameter re-estimation to the models.

From the context-independent character model set thus obtained, models for arbitrary words of some given lexicon can be constructed easily by concatenating the appropriate character models.



Sentence Database F01-062

It is short and familiar and its many meanings can be otherwise expressed only by cumbersome and abstract terms which then introduce new difficulties. But it is advisable, in any particular context, to narrow down its meaning so as to avoid confusion. "Mind" has often been contrasted with "matter" in such a way as to suggest that the two are somehow opposed and incompatible.

Name:

(a)

It is short and familiar and
its many meanings can be
otherwise expressed only by
cumbersome and abstract
terms which then introduce
new difficulties. But it is
advisable, in any particular

(b)

(c)

Fig. 8. Examples of the data-sets used: (a) image of a scanned document from the IAM-database, (b) video-document for the same text-prompt written on the whiteboard, and (c) text-detection results for the video document.

8.3 Language Model

For estimating character-based language models the transcriptions of the training data and for word-based models the original text prompts were used. The raw n-gram probability distributions were smoothed by applying absolute discounting (discounting factor $\beta = 1$) and backing-off (cf. e.g. [4]).

A major limitation for the performance of a word-based language model in our configuration of training and test data arises from the fact, that the texts belong to different categories covering widely differing topics. From the total of 2534 word forms appearing in the text prompts of the cross-validation data (categories E & F) more than 48% never appeared in the training texts (categories A – D). Additionally, writers sometimes used varying hyphenation which introduces unseen word fragments. In the whiteboard data 316 different word forms are used, more than 26% of which are not covered by the training set. Therefore, we decided to include in addition to the lexicon of the training data all those word forms in the overall recognition lexicon, which are necessary to describe the text prompts from which cross-validation and test set were generated. From this word list a small number of entries was eliminated, which contained characters not present in the training material. The resulting recognition lexicon consists of 7485 entries including punctuation and word fragments resulting from hyphenation. The percentage of out-of-vocabulary words for both cross-validation and test data is approximately 0.5%.

8.4 Model Decoding

If no statistical restrictions on the possible sequences of words or characters are imposed, i.e. if no language model is used, decoding of the HMMs can be achieved by standard Viterbi beam-search. However, the combined use of a writing and a language model requires additional effort during the recognition process. Otherwise, the search might not be able to find the solutions, which truly maximize the combined HMM and n-gram score. Therefore, when using a language model in the recognition process we apply an enhanced version of the time-synchronous recognizer proposed in [10]. The search spaces for HMM states and recognition hypotheses are established at different levels of abstraction. Only at the level of word or character hypotheses the HMM scores are combined with the n-gram probabilities provided by the language model. Thus it can be assured, that also long span restrictions as represented by e.g. 5-gram models can be combined correctly with the writing model and their predictive power can be fully exploited. This method for decoding a combined HMM and n-gram model is roughly equivalent to a time-synchronous rescoring of the HMM-based hypotheses with the language model [9]

9 Results & Discussion

In order to evaluate the proposed methods for video-based whiteboard reading we carried out several experiments on the test set described in section 8.1. Whenever possible the results obtained are compared to those achieved by an off-line recognition system on the cross-validation data.

9.1 Text detection

The precondition for whiteboard reading is to robustly detect the image regions of the handwriting. Therefore, we at first investigated the effectiveness of the method for text detection described in section 5. Using the 20 image sequences for testing consisting of 152 handwritten lines of text, it turned out that a total of 188 image regions have been detected. 173 of these regions are correctly detected text regions. In only 15 cases errors occurred due to noise or line segmentation errors caused by touching or heavily overlapping lines. The discrepancy of the total number of originally written lines (152) and the overall number of correctly detected text regions (173) is caused by the incremental processing strategy. Thus, we observed that in 21 cases portions of text lines have been detected repeatedly (see e.g. figure 8).

Additionally, we investigated whether the sequence of detected regions corresponds to the chronological order in which the text lines were written on the board. We found out that in 9 cases from the overall number of 173 text regions the chronological order was not correct.

Table 2. Word error rates (WER) achieved for a 7485-word lexicon with and without using a bi-gram language model.

	% WER / perplexity	
	none	2-gram
Cross-validation	43.9 / (7485)	28.3 / 757
Test (whiteboard)	47.8 / (7485)	28.9 / 645

9.2 Lexicon-based Recognition

For lexicon-based recognition of whiteboard texts we used a carefully defined lexicon containing 7485 word forms (see section 8.3). The results achieved are summarized in table 2. Without the use of any restrictions on the possible word sequences we obtain a word error rate of 47.8%. Clearly, such a figure would not be acceptable for an automatic transcription system. However, with some limited knowledge about the expected texts represented as a bi-gram language model this figure could be improved to 28.9%. This corresponds to a reduction of the error rate of approximately 40%. Due to the widely differing lexicons of training and test data the bigram model has a very high perplexity on both test and cross-validation set. For a well trained language model that could be estimated on text data *matching* the topics of the final application – i.e. the test texts – a substantially lower perplexity can be expected. Therefore, word-based recognition results on white-board data could easily be improved further for better matching training and test conditions.

9.3 Lexicon-free Recognition

Ultimately, any handwriting recognition system should be able to recognize text independently from a predefined list of possible words. For such lexicon-free recognition a least some expectation on the possible sequence of characters is required. Therefore, we estimated character-based language models with *n*-gram lengths ranging from two to five (see section 8.3). These models were then used in conjunction with the context-independent character HMMs during the recognition process. The results obtained are shown in table 3. Without the restriction of a language model a character error rate of 31.0% is obtained, i.e. roughly every third character – including white space – is misrecognized. However, when using the statistical restrictions on possible character sequences as represented by the character based language models this figure can be improved significantly. With a 5-gram model a character error rate as low as 19.0% can be achieved on the whiteboard data.

Though the mismatch of lexicons between training and test data is a severe limitation for word-based recognition it has an advantage for the judgment of the lexicon-free results. In principle long-span *n*-gram models could learn the training lexicon and, therefore, results obtained with such a model might not be truly lexicon-free. In our configuration, however, learning of the word forms found in the training texts has very limited effect on the cross-validation and test data (see also section 8.3). Therefore, the low character error rates achieved impressively demonstrate the capability of the *n*-gram models to capture mode general characteristics of the character sequences.

9.4 Video vs. Off-line Recognition

The comparison of the recognition results obtained on the whiteboard data and on the scanned documents used for cross-

Table 3. Character error rates (CER) achieved with different n-gram language models.

	% CER / perplexity				
	none	2	3	4	5
Cross-validation	29.2 / (75)	22.1 / 12.7	18.3 / 9.3	16.1 7.7	15.6 / 7.3
Test (whiteboard)	31.0 / (75)	25.9 / 12.0	22.0 / 8.5	20.1 / 6.9	19.0 / 6.5

validation clearly shows better performance on the later ones. However, the difference in recognition quality is relatively small when considering the widely different nature of the documents used. This evidence makes it obvious, that the methods used for text-detection, preprocessing and feature extraction are capable of compensating for the majority of distortion effects found in the video data.

10 Conclusion

We presented a system for automatic whiteboard reading based on visual input. It is characterized by an incremental processing strategy, i.e. the text lines are extracted as soon as they are visible in the image. The pre-processing and feature extraction methods applied generate a data representation which is to a certain extent robust against variations concerning the writing style and the reduced quality of the video-based data. Evaluation results on a writer independent task were presented for both lexicon-based and lexicon-free recognition of unconstrained handwriting. When using a 7.5k lexicon and a bi-gram model a word error rate of only 28.9% could be achieved. Without an explicit lexicon and the use of only a character 5-gram model a character error rate as low as 19.0% was reached. These results clearly demonstrate the effectiveness of the proposed methods text detection, preprocessing, feature extraction, and statistical modeling and recognition and their successful combination in a complete system for automatic video-based whiteboard reading.

Acknowledgements

This work was supported by the German Research Foundation (DFG) within project **Fi799/1**.

Additionally, we would like to thank the Institute of Informatics and Applied Mathematics, University of Bern, namely Horst Bunke and Urs-Viktor Marti, who allowed us to use the IAM database of handwritten forms [15] for our recognition experiments.

References

1. M. J. Black and A. D. Jepson. A probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions. In H. Burkhardt and B. Neumann, editors, *European Conf. on Computer Vision*, pages 909–924, Freiburg, Germany, 1998.
2. H. Bunke, T. von Siebenthal, T. Yamasaki, and M. Schenkel. Online handwriting data acquisition using a video camera. In *Proc. Int. Conf. on Document Analysis and Recognition*, pages 573–576, Bangalore, 1999.

3. R. G. Casey and E. Lecolinet. A survey of methods and strategies in character segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(7):690–706, 1996.
4. S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13:359–394, 1999.
5. P. Clark and M. Mirmehdi. Recognising text in real scenes. *Int. Journal on Document Analysis and Recognition*, 4:243–257, 2002.
6. J. G. A. Dolfing and R. Haeb-Umbach. Signal representations for Hidden Markov Model based on-line handwriting recognition. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, volume IV, pages 3385–3388, München, 1997.
7. S. Elrod, R. Bruce, R. Gold, D. Goldberg, F. Halasz, W. Janssen, D. Lee, K. McCall, E. Pedersen, K. Pier, J. Tang, and B. Welch. Liveboard: A large interactive display supporting group meetings, presentations and remote collaboration. In *Proceedings of ACM CHI'92 Conference*, pages 599–607, May 1992.
8. G. A. Fink. Developing HMM-based recognizers with ESMERALDA. In V. Matoušek, P. Mautner, J. Ocelíková, and P. Sojka, editors, *Lecture Notes in Artificial Intelligence*, volume 1692, pages 229–234, Berlin Heidelberg, 1999. Springer.
9. G. A. Fink and G. Sagerer. Zeitsynchrone Suche mit n-Gramm-Modellen höherer Ordnung. In *Konvens 2000 / Sprachkommunikation*, ITG-Fachbericht 161, pages 145–150. VDE Verlag, Berlin, 2000.
10. G. A. Fink, C. Schillo, F. Kummert, and G. Sagerer. Incremental speech recognition for multimodal interfaces. In *Proc. Annual Conference of the IEEE Industrial Electronics Society*, volume 4, pages 2012–2017, Aachen, 1998.
11. G. A. Fink, M. Wienecke, and G. Sagerer. Video-based online handwriting recognition. In *Proc. Int. Conf. on Document Analysis and Recognition*, pages 226–230, 2001.
12. S. Johannson, G. N. Leech, and H. Goodluck. *Manual of Information to Accompany the Lancaster-Oslo/Bergen Corpus of British English, for use with digital Computers*. Department of English, University of Oslo, Oslo, 1978.
13. H. Li, D. Doermann, and O. Kia. Automatic text detection and tracking in digital video. *IEEE Transactions on Image Processing*, 9(1):147–156, 2000.
14. U.-V. Marti and H. Bunke. Erkennung handgeschriebener Wortsequenzen. In P. Levi, R.-J. Ahlers, F. May, and M. Schanz, editors, *Mustererkennung 98, 20. DAGM-Symposium Stuttgart, Informatik aktuell*, pages 263–270, Berlin, 1998. Springer-Verlag.
15. U.-V. Marti and H. Bunke. A full english sentence database for off-line handwriting recognition. In *Proc. Int. Conf. on Document Analysis and Recognition*, pages 705–708, Bangalore, 1999.
16. U.-V. Marti and H. Bunke. Handwritten sentence recognition. In *Proc. Int. Conf. on Pattern Recognition*, volume 3, pages 467–470, Barcelona, 2000.
17. U.-V. Marti and H. Bunke. The IAM-database: An english sentence database for offline handwriting recognition. *Int. Journal on Document Analysis and Recognition*, 5:39–46, 2002.
18. M. Mirmehdi, P. Clark, and J. Lam. Extracting low resolution text with an actvie camera for ocr. In J. Sanchez and F. Pla, editors, *Proc. 9th Spanish Symposium on Pattern Recognition and Image Processing*, pages 43–48, 2001.
19. M. E. Munich and P. Perona. Visual input for pen-based computers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(3):313–328, March 2002.
20. N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. on Systems, Man, and Cybernetics*, 9:62–66, 1979.
21. R. Plamondon and S. N. Srihari. On-line and off-line handwriting recognition: A comprehensive survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(1):63–83, 2000.

22. E. Saund. Bringing the marks on a whiteboard to electronic life. In *Proc. 2nd Int. Workshop on Cooperative Buildings, CoBuild'99*, pages 69–78, Pittsburgh, 1999. Springer.

23. L. Schomaker and E. Segers. Finding features used in the human reading of cursive handwriting. *Int. Journal on Document Analysis and Recognition*, 2:13–18, 1999.

24. A. W. Senior and A. J. Robinson. An off-line cursive handwriting recognition system. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(3):309–321, 1998.

25. Q. Stafford-Fraser and P. Robinson. Brightboard: A video-augmented environment. In *Proc. Conf. on Human Factors and Computing Systems*, pages 134–141, Vancouver, BC, Canada, 1996.

26. T. Steinherz, E. Rivlin, and N. Intrator. Offline cursive script word recognition – A survey. *Int. Journal on Document Analysis and Recognition*, 2(2):90–110, 1999.

27. M. Wienecke, G. A. Fink, and G. Sagerer. A handwriting recognition system based on visual input. In *2nd International Workshop on Computer Vision Systems*, pages 63–72, Vancouver, Canada, 2001.

28. M. Wienecke, G. A. Fink, and G. Sagerer. Experiments in unconstrained offline handwritten text recognition. In *Proc. 8th Int. Workshop on Frontiers in Handwriting Recognition*, Ontario, Canada, August 2002.

29. Z. Zhang and C. Tan. Restoration of images scanned from thick bound documents. In *Proc. Int. Conf. on Image Processing*, pages 1074–1077, Thessaloniki, Greece, October 2001.