A Method for Camera-Based Interactive Whiteboard Reading

Szilárd Vajda, Leonard Rothacker, and Gernot A. Fink

TU Dortmund, Department of Computer Science,
Dortmund, Germany
{szilard.vajda,leonard.rothacker,gernot.fink}@udo.edu

Abstract. Recognizing mind maps written on a whiteboard is a challenging task due to the unconstrained handwritten text and the different graphical elements — i.e. lines, circles and arrows — available in a mind map. In this paper we propose a prototype system to recognize and visualize such mind maps written on whiteboards. After the image acquisition by a camera, a binarization process is performed, and the different connected components are extracted. Without presuming any prior knowledge about the document, its style, layout, etc., the analysis starts with connected components, labeling them as text, lines, circles or arrows based on a neural network classifier trained on some statistical features extracted from the components. Once the text patches are identified, word detection is performed, modeling the text patches by their gravity centers and grouping them into possible words by density based clustering. Finally, the grouped connected components are recognized by a Hidden Markov Model based recognizer. The paper also presents a software tool integrating all these processing stages, allowing a digital transcription of the mind map and the interaction between the user, the mind map, and the whiteboard.

Keywords: whiteboard reading; unconstrained document layout analysis; handwriting recognition

1 Introduction

Nowadays, in the field of handwriting recognition the focus is shifted from classical topics like bank checks or postal documents recognition [14] to more challenging topics like historical documents recognition, personal memos or sketch interpretation [15] and lately to recognition of unconstrained whiteboard notes [11, 13]. The later is in the focus of the attention because it deals with an unconstrained type of documents with no specific writing style, layout, etc.

Doing collaborative work (e.g. brainstormings, discussions, presentations) is quite common a in corporate or academical environment. However, there is just a limited amount of work [11, 13, 18] to embed this whiteboard outcome in a smart environment scenario (e.g. a conference room). To provide not just a digital capture of the whiteboard, but also the recognized content in an interactive software framework, is one of the final goals of such a smart room.

Fig. 1: Scene from a mindmap creation process around the idea of ”Whiteboard reading”.

Instead of tackling this issue by some specific (sometimes costly) hardware (e.g. special whiteboard, several cameras, pen, wireless microphone proposed by the e-Learning system [18]), we propose a system which uses only regular hardware (a simple whiteboard, markers, a low-resolution active camera and a projector) available in each conference room. Such hardware setup provides us a natural environment to actively support the collaborative mind mapping [3], allowing the users to keep their old habits and writing down their ideas using just the whiteboard markers without bothering about some special equipment. The focus is rather on the content and not on the layout. Such a mind map creation process is depicted in Fig. 1.

The current system focuses on two main aspects. First, we will present the system capable to recognize on-line the different text and non-text components and secondly, we will concentrate on the digital outcome of that recognition process: a digital, editable mind map framework and the interaction between the static whiteboard content, the user and the projected and already recognized mind map. Such an interaction is missing from the currently available systems. The scientific challenges lie in the facts that we analyze the documents without any prior knowledge, no curve tracing is considered, and due to the reduced number of such mind maps, the handwriting recognizer is trained on a completely different data.

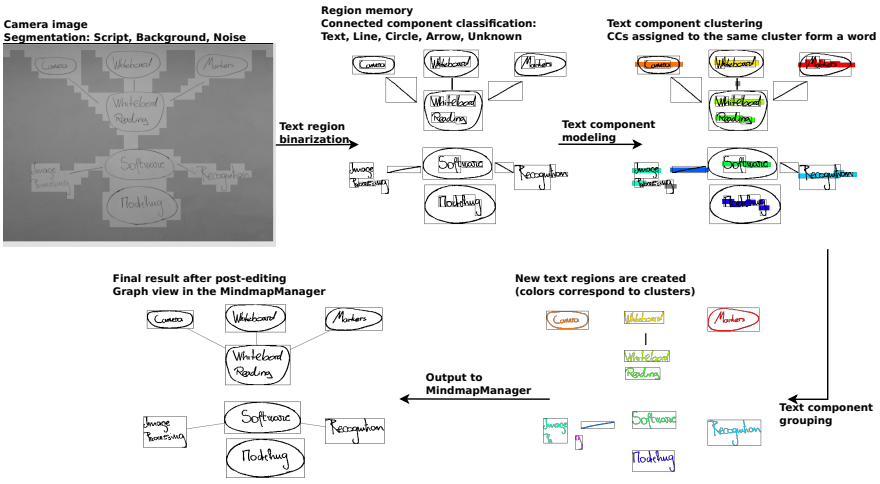


Fig. 2: Overview of the proposed whiteboard reading system.

The following sections of the paper are organized as follows. Related works concerning the whiteboard recognition will be discussed in the next section. Section 3 describes in detail the complete whiteboard reading system. Section 4 is completely dedicated to the data and the experimental setup. Finally, Section 5 summarizes and highlights the strengths of the presented reading system.

2 Related Work

Over the last two decades an impressive progress has been achieved in the field of handwriting recognition, even for large vocabularies [12] and multi-writer scenarios. However, the main constraint was always the same. To produce those sound results clean and well segmented data was necessary. In the whiteboard reading scenario addressed in this paper such presumptions can not hold. In particular, for collaborative works like mind mapping, different layouts, writers, styles and text and non-text mixture should be handled.

The very first attempt to handle such a challenging task was addressed by Wienecke et al. [16], where complete handwritten sentences were recognized using a low-resolution camera. The proposed prototype system was quite promising, however it was able to recognize text only.

A similar type of research was conducted in [7, 8], where the data acquisition was performed on-line using an infrared sensor for tracking a special pen. Even though the results are sound, it to be noted, that only clear, well structured text lines were recognized.

To recognize Japanese characters on a whiteboard, the authors in [19] consider a complex hardware scenario including two cameras and a special pen to capture the

writing. The text detection is not anymore performed by the system but rather by the software provided by the pen manufacturer. The system is used in an e-Learning scenario.

In a more recent work [11] of ours, we focused on a similar task, recognizing well-structured handwritten whiteboard paragraphs, considering only a camera and no on-line information. The results are really promising, however, the text detection on the whiteboard is based on some connected component (CC) estimation which is very rigid due to the usage of some thresholds.

Finally, in our recent work [13] we addressed the mind map recognition scenario (see Fig. 1), where beside the text components graphical elements like lines, circles, arrows were detected and a totally unconstrained document layout was analyzed. The goal of the current work is to improve that system by adapting the recognition to the different changing layouts, reconstruct the mind map and introduce a certain type of interaction between user, document and whiteboard.

3 Whiteboard reading system

In this section we concentrate on the whiteboard reading system, describing the complete process starting from the image acquisition, throughout the different processing steps and finally the recognition and the user interaction with the system. A system overview with its particular processing stages is shown in Fig. 2.

3.1 Image acquisition and camera-projector calibration

For image acquisition a camera and for user interaction a projector must be directed to the whiteboard. The camera is capturing the whole mind map creation process. Low-resolution gray level camera images are used for further processing (see Section 3.2).

The camera-projector calibration is needed to project content to the whiteboard that is derived from the camera image. In order to project information on the whiteboard for user interaction (see Section 3.7), a mapping between the camera- and the projection image coordinate systems has to be obtained. The projected image contains additional user information and is shown on the whiteboard using the projector. This way the projection image can be seen as an overlay to the mind map drawn with a marker by the user.

For calibration a chessboard is projected on the whiteboard that is captured with the camera. Because the chessboard is rendered in the projection image, its chessboard corner coordinates are known in the projection image coordinate system. By finding chessboard corners in the camera image correspondences between both images are obtained. Finally, a homography can be estimated that maps each point from the camera coordinate system to the projection image coordinate system. The chessboard corner localization and homography estimation that we use are both freely available [1].

3.2 Image segmentation

The purpose of image segmentation is to separate elements written on the whiteboard with a marker from the whiteboard background and noisy image parts. Noisy parts are

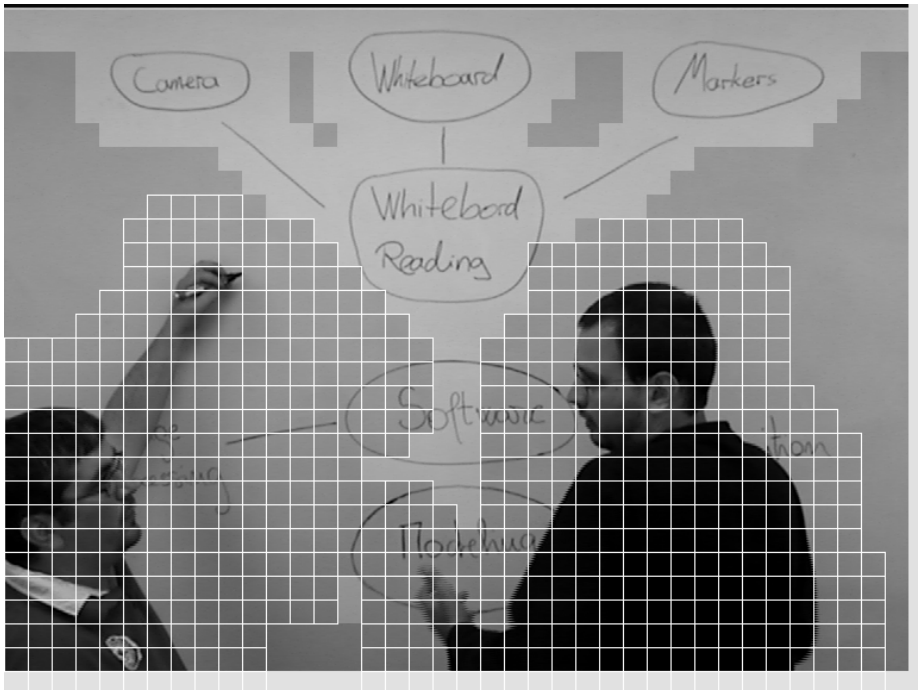


Fig. 3: The segmentation of the Fig. 1 into text, background and noise

for example regions where the user stands (see Fig. 1). Afterwards the regions containing written content are further segmented by categorizing them into different mind map elements (text, line, circle, arrow).

Segmentation of the camera image After image acquisition the objective is to extract only the content written on the whiteboard. This relevant information is then added to a binary region memory (also refer to Fig. 1). The region memory represents the current state of written content on the whiteboard and is robust to irrelevant changes in the camera image, like illumination or particular users standing in front of the whiteboard. Therefore the general assumption is that the camera image does not contain anything but the interior of the whiteboard. The camera and the whiteboard are fixed. In this scenario the system has to handle images that can consist of three different regions, namely:

- text (indicated by bright blocks in Fig. 3).
- background (indicated by dark blocks in Fig. 3).
- noise (indicated by blocks with grid pattern in Fig. 3).

As proposed by [16], segmentation is not done on pixel but on block level. The image is therefore divided into two layers of overlapping blocks. Each block is now segmented into one of the formerly mentioned categories on the basis of three features: gradients, gray level and changes between two consecutive images.

The first feature (ξ^k_{edge}) uses image gradients: A block breaks the edge threshold (Θ_{edge}) if it exceeds a number of gradients of a minimum magnitude. Gradient magnitudes are derived by convolving the block with horizontal and vertical Sobel masks. The minimum magnitude is an additional parameter that has to be given.

The second feature (ξ^k_{gray}) uses the average gray level: A block breaks the gray level threshold (Θ_{gray}) if its average gray level falls beyond the scaled overall image average gray level. That means that the block is darker than the overall image.

The third feature (ξ^k_{diff}) uses the change between two consecutive images: A block breaks the difference threshold (Θ_{diff}) if the sum-of-absolute-differences error metric computed between corresponding blocks exceeds the threshold value.

The categorization is done depending on whether the blocks meet the following criteria:

- Text:
$$\left(\xi^k_{\text{edge}} > \theta_{\text{edge}}\right) \wedge \left(\xi^k_{\text{gray}} > \theta_{\text{gray}}\right) \wedge \left(\xi^k_{\text{diff}} < \theta_{\text{diff}}\right).$$
Text blocks contain many strong gradients and have a bright average gray level because the pen stroke is the only dark element on a bright background. Furthermore, there should be no movement since both camera and whiteboard are supposed to stand still.
- Noise:
$$\left(\xi^k_{\text{gray}} \leq \theta_{\text{gray}}\right) \vee \left(\xi^k_{\text{diff}} \geq \theta_{\text{diff}}\right).$$
Noise blocks are mainly caused by users being captured by the camera. In contrast to the whiteboard their appearance is generally darker. Additionally, their main activity will be writing to the whiteboard, so we can assume that blocks containing users also contain movement.
- Background: If the block is considered neither text nor noise. The camera is supposed to capture only the interior of the whiteboard, thus a block can be considered background if it is not text or noise.

After categorizing all blocks the region memory can be updated.

Noise blocks are discarded because the whiteboard is potentially occluded at those locations. To be even more robust also blocks in a noise block's local neighborhood can be discarded. The occurrence of eventually appearing parts of the user's shape in the region memory can be minimized this way. The information contained in a falsely discarded block will simply be added to the region memory later.

Background blocks do not contain any written content, so the corresponding regions in the region memory can be erased.

Finally text blocks are binarized with the local Niblack method [10] and inserted into the region memory if their XOR errors with the region memory exceed a certain empirically selected threshold. This way the memory does not get updated for very small changes in the camera image but only if there is a modification to the written content. Those small changes are likely to be caused by illumination changes in the conference room. The different thresholds considered in the segmentation process were selected based on trial runs. Though the detection is stable, considerable change in lighting conditions requires a new threshold set. For further details please refer to [16].

The result as depicted in Fig. 2 consists of a binary representation of the whiteboard content and can be used for further processing.

Segmentation of the whiteboard image A key issue to success is the accurate segmentation of the whiteboard content. We separate the whiteboard from the rest of the scene (see Section 3.2), but we do not have any prior information about the content itself. To recognize and reconstruct the mind map, we need to separate text elements from non-text items, namely in this scenario, lines, circles and arrows. The detection process is based on connected components (CC) extracted from the binarized image. Working with CC is suitable as the connected components are easy to extract and no specific prior knowledge is necessary.

Instead of using heuristics - rooting from the work of Fletcher and Kasturi [5], we propose a solution to classify CCs based on statistical learning. A descriptor of 12 components (i.e. contrast, edge density, homogeneity, number of foreground gray levels, foreground mean gray level, relative amount of gradient orientations, Sobel gradient orientation and magnitude, etc.) is extracted from each CC and a multi-layer perceptron is meant to classify the pixel patches into text, line circle and arrow. For more details, please refer to [13]. This text component detector is suitable not only for Roman script but also for Chinese, Arabic or Bangla, where even more complex characters shapes will occur.

3.3 Layout analysis

The layout analysis of a document consists of identifying the baseline elements composing the document and their spatial relationship among each other. While for printed documents a certain type of regularities like font type, font size, line structures, etc. can be detected, in a handwritten mind map documents none of these is to be identified, hence the layout analysis is more challenging in such unconstrained handwritten document scenarios.

Layout modeling As described above, we separate first text items from non-text items. For further processing we will concentrate our effort to model only the different text patches. The lines, circles and arrows detected previously will serve to build the digital representation of the mind map into the so-called ”MindMap Manager”, discussed later in Section 3.6.

Our proposition is to adapt the model to the analyzed document considering the gravity centers of the text CCs (see Fig. 2a) and model the structure of the text patches (CCs) throughout these points. For each text component, the gravity center is calculated. At the left and right side of the bounding box a new center is calculated inheriting the height form the original gravity center. For larger components, exceeding the *average width*, estimated over all connected components from the document, at each slice of *average width*/4 of the CC, a new gravity center is computed w.r.t. the pixels counted in that window.

3.4 Word detection

Once the modeling part is done, the different gravity centers will form ”dense” regions (see Fig. 2) corresponding to possible words. These agglomerations into different clusters

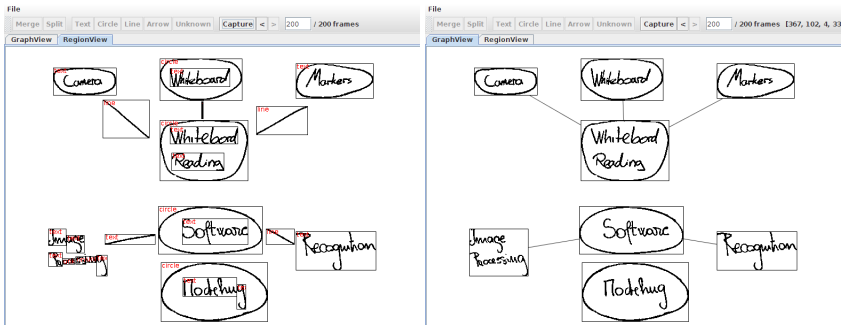


Fig. 4: User interface of the *Mindmap Manager* showing the region view on the left and the graph view on the right. In the region view segmentation and recognition results can be corrected. The graph view shows the final graph representation of the mind map.

need to be identified in order to separate the different words from each other. For this purpose the DBSCAN algorithm [2] has been considered. While other clustering methods rely mainly on some distance metrics, in this case the distance is combined with the density.

The gravity centers will be clustered not only by the distances (between the different text patches), but also by the density which is definitely higher around the different text components (see Fig. 2).

Let $D_n = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ be the coordinates of the gravity centers, where $x_i, y_i \in R^+$ and n denotes the number of gravity centers in the set.

Let us denote by β-neighborhood of a point $p_k \in D_n$ the $N_\beta(p_k) = \{(p_k) \in D_n | dist(p_k, p_l) < \beta, k \neq l\}$, where $dist()$ is the Euclidean distance.

Considering the β-neighborhood of a point, we define the notion of: p_k is density reachable from p_l if $p_k \in N_\beta(p_l)$ and $|N_\beta(p_l)| \geq P_{min}$, where P_{min} is the minimal number of points (gravity centers) which should be around point p_k.

The proposed clustering process is based on the β-neighborhood of a given point. We select as belonging to one cluster all the points which are density reachable considering a given number of k (number of neighbors). The expansion of each cluster is based on this idea allowing to get rid of the noisy points which density reachability indices are lower than for the others. For more details about the clusters expansion, please refer to work [2]. Finally, the original CCs' gravity centers are mapped to the different clusters established by DBSCAN (see Fig. 2).

3.5 Word recognition

For the recognition, we use the same recognizer that in our previous work was successfully applied to the task of reading text paragraphs in high-resolution whiteboard images [11]. Handwritten words are modeled by semi-continuous character HMMs. A sliding window is applied on the normalized text snippets considering 8 pixel wide analysis window with 25% overlap. For each frame a set of nine geometric features and

the approximation of their first derivatives are computed [17]. In total, 75 models considering upper and lower case letters, numerals and punctuation marks have been trained on the IAM database [11]. The HMM models are decoded with a time-synchronous Viterbi beam search algorithm [4].

3.6 The Mindmap Manager

The *Mindmap Manager* is the front end of the presented system. Segmentation, grouping and recognition results are consecutively — for each change in the region memory — saved and made accessible to the user for post-editing and exporting. Fig. 4 shows its user interface. The region view (left side of Fig. 4) contains editing functionality for incorrectly segmented or classified components. Please note that those components refer to classified CCs that in case of text elements might have undergone further grouping. After selecting a component its category can be changed or the entire component can be split horizontally or vertically. If the categories of two different components are compatible, they can be merged.

The graph view (right side of Fig. 4) contains a graph representation of the mind map. Text and circle elements are treated as nodes and lines and arrows are treated as edges (compare region and graph view in Fig. 4). This way the user is able to rearrange the layout of the nodes by simply dragging and dropping them. Connected edges will follow. Besides the possibility to rearrange the mind map layout the graph view has no further editing capabilities and is rather intended to contain the final outcome of the system. A compatibility with existing digital mind map formats would allow to use other, already existing tools to further manipulate the documents.

The graph representation has to be created from the previously mentioned components. Because there is no prior knowledge of which nodes are to be connected by a line, an estimation is necessary. By transferring components classified as lines to Hough space [6], a parametric representation of the line can be estimated. Finally, two nodes being connected by the line component are determined trough intersection of the parametric line with neighboring components.

3.7 Interaction with the whiteboard

The purpose of user interaction is to give a feedback of segmentation and recognition results while the mind map creation is still in progress. This way the user can react accordingly (e.g. writing things clearer) to improve the overall performance. The feedback is given by highlighting newly recognized elements on the whiteboard using the projector. The highlighting color indicates the classification result (text, line, circle, arrow). To use the camera and the projector in conjunction a camera-projector calibration has to be performed initially (also see Section 3.1).

Segmentation and recognition results can be retrieved whenever the region memory changes (see Section 3.2). Ideally those updates to the region memory occur only if there is a change to the written content on the whiteboard. In such cases changed CCs are determined by computing a difference (XOR) image in the region memory. From their bounding boxes highlighting events are now generated that additionally contain the category (text, line, circle, arrow) for highlighting in a specific color. The bounding

boxes are given in camera image coordinates and have to be mapped to projection image coordinates for rendering. This mapping can be computed through the formerly estimated homography (see Section 3.1). After rendering, the user will see a colored rectangle around all recently changed parts of the mind map for a few seconds. In order to be more robust to false updates of the region memory (e.g. due to illumination changes), highlighting events will only be generated if the change in the region memory exceeds a certain threshold.

Finally we have to deal with the effect that projections to the whiteboard are also captured by the camera. This way projections can result in CCs being extracted that do not correspond to written content on the whiteboard. The idea is to filter the area in the camera image, where there will be a projection, from the region memory.

4 Experiments

In this section a brief description of the data and the results achieved by the described method will be presented.

4.1 Data description

The dataset consist of 31 mind maps written by 11 different writers around the topics "study", "party" and "holiday". 2 writers sketched only 2 mind maps. All writers were provided with different color markers. While the usage of some words was imposed, the writers were not restricted in creating their own mind maps around the previously mentioned topics. Once the mind map was ready a photo (2048x1536 resolution) of the whiteboard was taken. The data is annotated with respect to the nature of connected components (line, circle, text, arrow) and words [13].

4.2 Results

To evaluate thoroughly the method we need to evaluate the text detection solution, the subsequent modeling strategy and the text recognition. As the text detection method was originally proposed in [13], we just briefly describe the results, and we focus rather our evaluation on the layout analysis and the recognition. For more details on the results concerning the text detection, please refer to [13].

Text Detection: The neural network provides an average recognition score of 95.7% for the different CCs as being text, line, circle or arrow. However, while for text components the recognition scores are high (99.4%), for lines and arrows there are elevated confusion rates.

Layout Analysis: For the evaluation of the proposed method we use the method introduced in the context of the ICDAR 2005 Text Locating Competition [9]. That way we produce comparable and comprehensible evaluation results. The bounding boxes of the annotated ground truth T and the agglomerated text components E are compared – the larger the overlap of the bounding boxes, the higher the level of match. A match m_p

between two rectangles r, r' is defined as the quotient of their intersection area and their union area:

$$m_p = \frac{A(\bigcap(r, r'))}{A(\bigcup(r, r'))}. \tag{1}$$

The evaluation scheme is based on *precision* and *recall*. Having a binary answer to whether there is a fitting ground-truth rectangle to an estimated one or not would not cope with partial matches. This is why the quality for a single match m_p in this case lies in the range of $[0; 1]$. In order to calculate these adapted versions of precision and recall the best match between a rectangle within the agglomerations and all rectangles within the set of annotations is taken into consideration – and vice versa. The best match $m(r, R)$ of a rectangle r within a set of other rectangles R is defined as:

$$m(r, R) = \max \{m_p(r, r')|r' \in R\}. \tag{2}$$

The *recall* then is the quotient of the sum of the best matches of the ground truth among the agglomerated areas and the number of all annotated bounding boxes within the ground truth.

$$recall = \frac{\sum_{r_t \in T} m(r_t, E)}{|T|}. \tag{3}$$

The *precision* relates to the quotient of the sum of the best matches of the agglomerated areas among the annotated regions and the number of all agglomerated areas:

$$precision = \frac{\sum_{r_e \in E} m(r_e, T)}{|E|}. \tag{4}$$

We evaluated the output of the agglomeration (modeling) using both schemes described above. In Fig. 5 we display a typical result of the hierarchical clustering, stating in this case the maxima for precision and recall at 75% and 72%, respectively. The average recall value for the test documents is 67.09%. The main error source is due to the high number of non-text patches labeled as text not retrieved anymore in the ground truth.

While in some cases, the agglomeration is successful, in some other cases it fails because of some CCs were recognized as non-text (e.g. M in “Motto” or D in “Dance” in Fig. 7) or due to some distances which lead to agglomeration or separation (see “Guests” in Fig. 7) of different text items. Overall, in 211 cases the agglomeration produced non-text word hypothesis, in 194 cases some parts of the word (mainly characters) were missing. Finally, in 353 cases complete words, or words preceded or followed by noises (mainly lines) were detected. Some typical agglomeration errors are depicted in Fig. 6. Also a common error is encountered, namely the first letters of the words are often are connected with the surrounding circles, hence the letter is analyzed with the circle and classified as graphical element (e.g. F in ”Food” or T and l in ”Tunnel” in Fig. 7). This is one major limitation of the connected component based methods. To separate these elements, more sophisticated methods like skeletonization and curves tracing solutions should be applied.

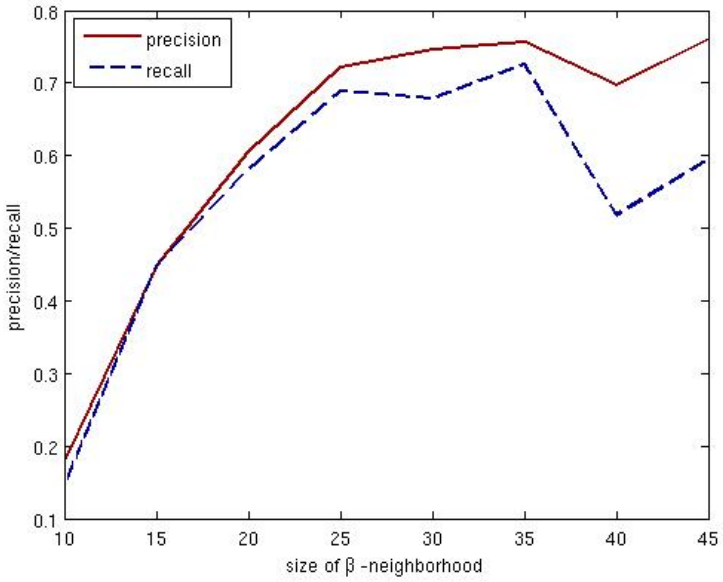
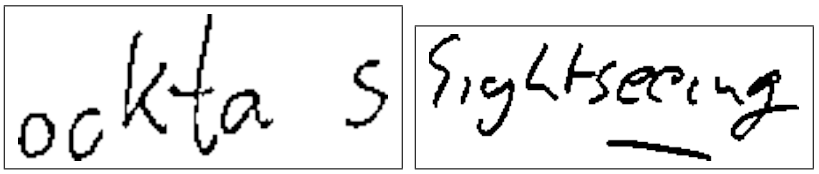


Fig. 5: Layout analysis results (Precision and recall) for the document shown in Fig. 7.



(a) missing letters (b) agglomeration with noise

Fig. 6: Typical error cases occurred in the agglomeration

Word Recognition: The recognition of the word snippets by the HMM are reported only on those 353 words considered as successful (complete) for the grouping. The lexicon (164 entries) was generated from the transcripts including all tentatively written words irrespective of segmentation errors. To reject test fragments with erroneous ink elements a rejection module was used, defined as an arbitrary sequence of character models. The overall word recognition score in the different snippets is 40.5%. 83.3% of the snippets were recognized correctly (i.e one word snippets). The low scores can be explained by the fact that the recognizer is trained on completely different data, while the recognition is performed on low-resolution image snippets, with huge writing style variations and containing also additional noise components inherited from the grouping process.

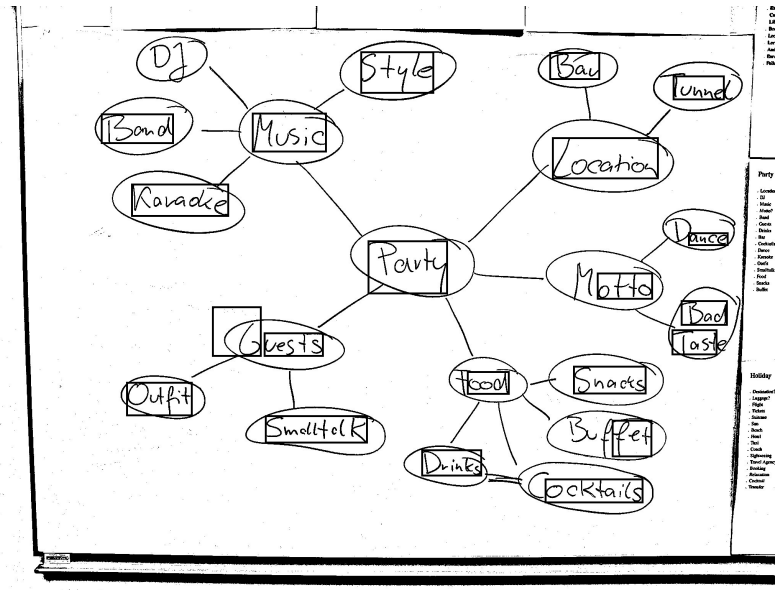


Fig. 7: Layout analysis results on an exemplary mind map.

5 Summary

In this paper we proposed a basic prototype reading system to automatically recognize mind maps written in an unconstrained manner on a whiteboard. Instead of considering expensive equipment, only common tools like e.g. whiteboard, markers, camera, and a projector were considered in the recognition scenario, usually available items in a conference room.

Instead of establishing some rules, the method adapts to the layout of each analyzed document. The modeling of the text components by their gravity centers followed by Density Based Spatial Clustering will provide the solution to merge the detected text patches (connected components) into words which serve as input for a handwriting recognizer. For this preliminary work, the recognition results, even though the recognizer was trained on completely different data, are not satisfying yet, but with some post-processing of the grouping more complete and accurate word agglomerations can be submitted to the recognizer. The software tool and the interactivity with the whiteboard provides a straightforward solution for a human-computer interaction in this challenging automatic whiteboard reading scenario.

Acknowledgment

This work has been supported by the German Research Foundation (DFG) within project **Fi799/3**.

References

1. http://opencv.willowgarage.com/wiki/, openCV (Open Source Computer Vision) library
2. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: International Conference on Knowledge Discovery and Data Mining. pp. 226–231 (1996)
3. Farrand, P., Hussain, F., Henessy, E.: The efficiency of the "mind map" study technique. Journal of Medical Education 36(5), 426–431 (2003)
4. Fink, G.A.: Markov Models for Pattern Recognition, From Theory to Applications. Springer, Heidelberg (2008)
5. Fletcher, L., Kasturi, R.: A robust algorithm for text string separation from mixed text/graphics images. IEEE Trans. on Pattern Analysis and Machine Intelligence 10(6), 910–918 (November 1988)
6. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edn. (2001)
7. Liwicki, M., Bunke, H.: Handwriting recognition of whiteboard notes. In: Conference of the International Graphonomics Society. pp. 118–122 (2005)
8. Liwicki, M., Bunke, H.: Handwriting recognition of whiteboard notes – studying the influence of training set size and type. International Journal of Pattern Recognition and Artificial Intelligence 21(1), 83–98 (2007)
9. Lucas, S.M.: Text locating competition results. In: International Conference on Document Analysis and Recognition. pp. 80–85 (2005)
10. Niblack, W.: An introduction to digital image processing. Strandberg Publishing Company, Birkeroed, Denmark (1985)
11. Plötz, T., Thurau, C., Fink, G.A.: Camera-based whiteboard reading: New approaches to a challenging task. In: International Conference on Frontiers in Handwriting Recognition. pp. 385–390 (2008)
12. Plötz, T., Fink, G.A.: Markov models for offline handwriting recognition: a survey. International Journal on Document Analysis and Recognition 12(4), 269–298 (2009)
13. Vajda, S., Plötz, T., Fink, G.A.: Layout analysis for camera-based whiteboard notes. Journal of Universal Computer Science 15(18), 3307–3324 (2009)
14. Vajda, S., Roy, K., Pal, U., Chaudhuri, B.B., Belaid, A.: Automation of Indian postal documents written in Bangla and English,. International Journal of Pattern Recognition and Artificial Intelligence 23(8), 1599–1632 (2009)
15. Weber, M., Eichenberger-Liwicki, M., Dengel, A.: a.scatch - a sketch-based retrieval for architectural floor plans. In: International Conference on Frontiers of Handwriting Recognition. pp. 289–294 (2010)
16. Wienecke, M., Fink, G.A., Sagerer, G.: Towards automatic video-based whiteboard reading. In: International Conference on Document Analysis and Recognition. pp. 87–91. Washington, DC, USA (2003)
17. Wienecke, M., Fink, G.A., Sagerer, G.: Toward automatic video-based whiteboard reading. International Journal on Document Analysis and Recognition 7(2-3), 188–200 (2005)
18. Yoshida, D., Tsuruoka, S., Kawanaka, H., Shinogi, T.: Keywords recognition of handwritten character string on whiteboard using word dictionary for e-learning. In: Proceedings of the 2006 International Conference on Hybrid Information Technology - Volume 01. pp. 140–145. ICHIT '06, IEEE Computer Society, Washington, DC, USA (2006)
19. Yoshida, D., Tsuruoka, S., Kawanaka, H., Shinogi, T.: Keywords recognition of handwritten character string on whiteboard using word dictionary for e-learning. In: International Conference on Hybrid Information Technology. pp. 140–145 (2006)