Camera-Based Analysis of Whiteboard Notes

Szilárd Vajda, Tobias Ramforth
TU Dortmund University
Department of Computer Science XII
44221 Dortmund, Germany
{Szilard.Vajda,Tobias.Ramforth}@udo.edu

Thomas Plötz, Gernot A. Fink
TU Dortmund University
Robotics Research Institute
44221 Dortmund, Germany
{Thomas.Ploetz,Gernot.Fink}@udo.edu

Abstract

A domain where, even in the era of electronic document processing, handwriting is still widely used is note-taking on a whiteboard. Such documents are either captured by a pen-tracking device or – which is much more challenging – by a camera. In both cases the layout analysis of realistic whiteboard notes is an open research problem.

In this paper we propose a camera-based three-stage approach for the automatic analysis of whiteboard documents. Assuming a reasonable foreground-background separation of the handwriting it starts with a locally adaptive binarization followed by connected component extraction. These are then automatically classified as representing either simple graphical elements of a mindmap or elementary text patches. In the final stage the text patches are subject to a clustering procedure in order to generate hypotheses for those image regions where textual annotations of the mindmap can be found.

In order to demonstrate the effectiveness of the proposed approach we report results of an experimental evaluation on a data set of mindmap images created by several different writers without any constraints on writing or drawing style.

1. Introduction

In many areas writing down notes or texts manually using, for example, pens has been replaced by machine-based techniques. Very prominently, it is nowadays standard to write an email using a computer and a keyboard rather than actually writing a letter. Without any doubts, electronically supported creation of documents implies several advantages. Machine-printed texts are easily to read by virtually everybody. Furthermore, storage and retrieval are more convenient for electronic rather than for handwritten documents.

However, there are still certain application cases where the "traditional" way of handwriting is more favorable [16].

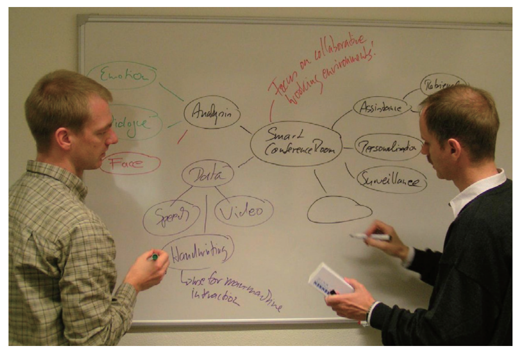


Figure 1. Mindmap creation on a whiteboard

Especially for creative processes like brainstorming any (electronic) equipment that might distract the attention of humans is likely to hinder the process of generating ideas. Basically, distraction kills creativity. Consequently, in such cases people often fall back on "low-tech" equipment for writing down their ideas, namely to pens and paper.

A standard means of writing down the results of a brainstorming session in a well structured way is *mindmapping* [1]. A mindmap basically corresponds to a graph with nodes and edges. Nodes represent the ideas that are usually written down as short texts – mostly a single or just a small number of words each. Relations between certain ideas are visualized by (directed) edges between these nodes. Apart from that, there is no constraint in how to organize a mindmap, for example, w.r.t. writing style, writing direction etc. For group based brainstorming mindmaps are usually created on a whiteboard, which is nowadays standard equipment of a meeting room. In Fig. 1 the collaborative creation of a mindmap is shown.

When restricting note taking to the use of pens and whiteboard, unfortunately, all advantages of electronically supported techniques (see above) are explicitly left out. However, particularly for storage and retrieval, digital representations of whiteboard notes in general and especially mindmaps written on it are desirable. For their creation the paradigm of non-obtrusiveness remains, though.

In our work we develop a camera-based automatic whiteboard reading system [13]. One goal is to monitor the dynamic process of creating mindmaps on a whiteboard using a video camera and to automatically extract a digital representation of the mindmap. The latter then can be used for the desired electronic storage and retrieval. By means of a projector recognized mindmaps can easily be reproduced directly at the whiteboard. This allows intuitive interaction (editing, erasing, browsing etc.) with the mindmap using natural means, i.e. pens, eraser and whiteboard.

One prerequisite for its successful recognition is the segmentation of a mindmap w.r.t. graphical elements (circles, lines, arrows) and text blocks. In this paper we present an approach for the automatic analysis of the structure of handwritten mindmap drawings. Still images of mindmaps written on whiteboards serve as input data. In a three-stage procedure we first extract relevant connected components, which are then fed into a classification system. At this second stage of the proposed procedure features calculated from the extracted connected components are automatically classified as either belonging to some graphical element or as being part of handwritten text. For a successful mindmap recognition we then agglomerate connected components of the same type to larger portions of structurally connected basic elements. Clusters of connected text components form single words that are the input for our handwriting recognition system. The output of the presented approach is a full segmentation of a mindmap image that includes region-based annotation at the level of graphical elements – circles, lines, arrows – and words. By means of an experimental evaluation on a database of mindmaps that have been sketched on whiteboards by multiple writers we demonstrate the effectiveness of this new approach.

2. Related Work

A digital document consists of a huge variety of physical items such as text blocks, lines, words, figures, tables and background, etc. However, at a lower level all these items are composed just by connected components, which are a set of interconnected pixels containing no high-level information at all. The goal of document structure and layout analysis is to detect the different regions and to identify the functional roles and relationships between them [9].

While a human reader uses several clues like context, and a-priori information about the script together with a complex reasoning mechanism, the machine can rely only on the extracted low-level information. This is the reason why automatic layout and structure analysis of an arbitrary document is a very challenging task. However, we should distinguish between printed documents and handwritten ones. While for printed documents we can presume a certain layout, structure [4] or textual information, like font size, boldness [6], for handwritten documents there is usually a total lack of physical organization.

While some impressive results have been achieved for the recognition of handwritten forms, postal documents [2, 14] and mathematical formulas [3, 11, 17], the analysis and recognition of whiteboard notes is a relatively new issue in the scientific community and just some attempts can be found in this subject.

In [7] the authors propose a system to recognize whiteboard notes by using an HMM based recognizer. In this system the image acquisition is performed on-line utilizing an infrared sensor. Unfortunately the work addresses just the problem of word recognition of well structured handwritten notes without considering any extra information, which can occur in such a document.

A kind of e-Learning strategy using a whiteboard has been described in [20]. The authors use two cameras and a pen capture tool on the whiteboard to recognize Japanese characters based on some character matching. However, to detect the text regions from the whiteboard, they consider the software provided by the pen manufacturer.

In [12] the task of processing whiteboard images is addressed using portable digital cameras or cell-phones. However, the work is more related to image processing rather than to its analysis. The focus is on the detection of board boundaries and on image quality enhancement. The output of the described procedure can then be used for further analysis.

The authors in [13] consider a more challenging issue as they recognize whiteboard notes taken with a camera and without any on-line information. Their text detection strategy is based on the different pieces of low-level information extracted from the connected components and all this is calculated as a probability. The drawback of this strategy is its rigidity as it considers global thresholds to distinguish between textual and non-textual items.

Back in the 1980s research around textual documents has been extended to line drawings. The original raw data was scanned documents but the aim was not to recognize the structure/layout and content but to rebuild the high-level design from engineering drawings, recognize pipes, lines, roads, rivers in maps, etc. [18]. Considering the content of these documents, maybe they are more complex than printed materials but still operating with a limited and well defined set of graphical items.

3. Camera-Based Segmentation of Mindmaps

A mandatory pre-processing step for successful recognition of hand-drawn mindmaps is their segmentation. The goal of this process is to annotate regions of a camera image w.r.t. graphical elements and text. We developed a three-

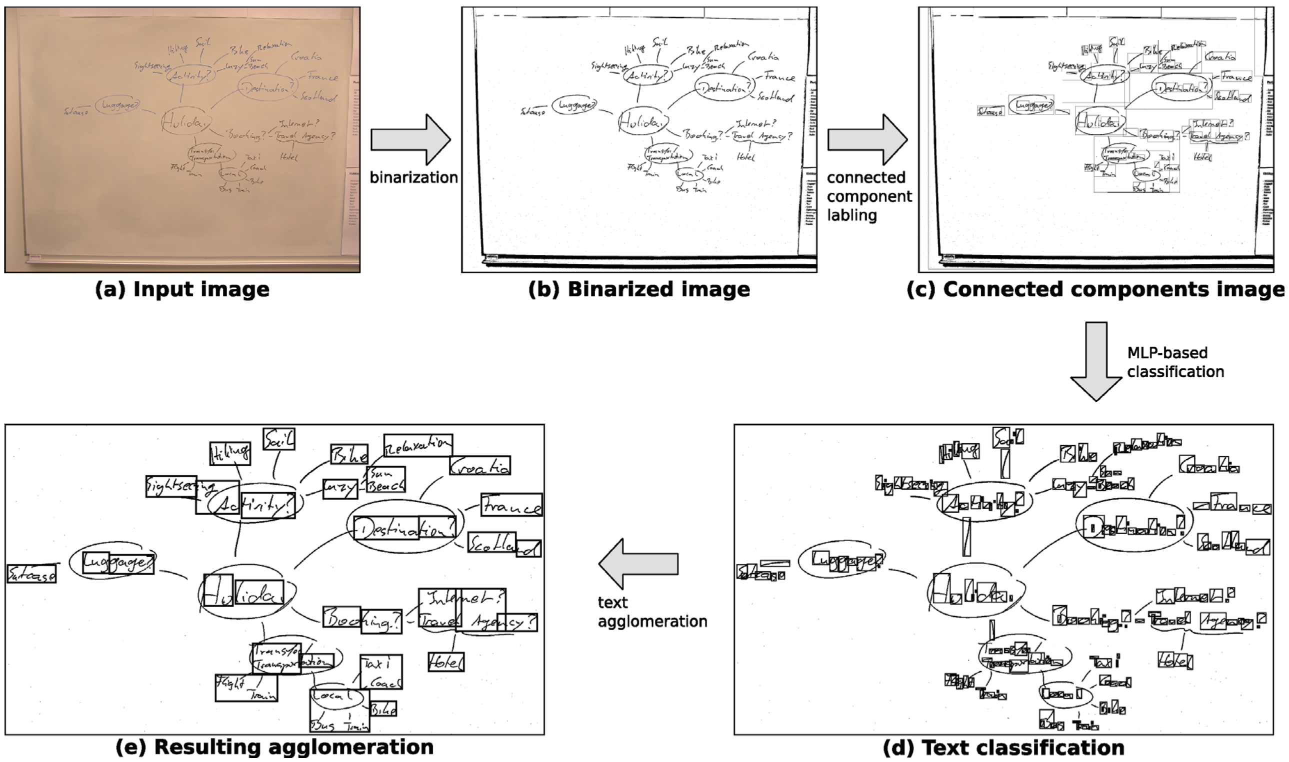


Figure 2. Overview of the system for automatic segmentation of whiteboard notes

stage procedure that handles still images of mindmaps and produces a complete region-based annotation. The overall procedure is illustrated in Fig. 2. It starts with the extraction of relevant connected components (cf. Fig. 2c). In the second stage, the latter are automatically classified using a statistical modeling approach. Therefore, feature representations of all connected components are fed into a classifier that provides a labeling w.r.t. circles, lines, arrows and text (cf. Fig. 2d). In the last stage connected components of the same type (graphical elements or text) are agglomerated by means of a hierarchical clustering procedure (cf. Fig. 2e). The output of the segmentation procedure, in terms of separate regions containing text blocks at word level as well as the other graphical elements, serves as input for a recognition system. It will analyze the graph structure and recognize handwritten notes.

3.1. Connected Component Extraction

In the first step of our segmentation method connected components are extracted. The structure of mindmap images suggests this procedure as there is apparently a clear distinction between handwriting in the foreground and a more or less homogeneous background (the surface of the whiteboard). Thus, connected components are very likely to be concentrated on the actual mindmap. Disregarding probable flaws in the image (e.g. inhomogeneous lighting, or non-opaque marker color) separating the mindmap by connected component analysis is reasonable.

For the purpose of connected component extraction the input image has to be binarized, which in our case is accomplished by use of Niblack's algorithm [10]. A variant of the basic approach is used that applies threshold optimization [15] and local thresholding in a 51x51 pixels window. For efficient computation integral images for plain as well as for squared pixel values are analyzed [19]. The actual extraction of connected components follows a straightforward approach of segmenting contiguous black pixel regions.

By means of heuristic post-processing connected components that obviously do not belong to the mindmap are suppressed by trivial filtering. The remaining set of connected components is not necessarily limited to well isolated, known graphical elements and text portions only. Instead unknown and touching elements together with additional clutter are very likely to occur (cf. Fig. 2c). Thus, for successful segmentation of mindmap images further analysis of the extracted connected components is required.

3.2. Classification of Connected Components

In the second stage of our segmentation approach the set of extracted connected components is classified w.r.t belonging to either one of the known graphical elements, text, or unknown. In the latter case the particular connected component is discarded from further processing.

In a mindmap circles/ellipses, lines and arrows are used for the purpose of grouping, linking and structuring. This stands in contrast to textual items which represent the actual content of a mindmap. Basically, all considered elements exhibit certain structural specialties. Textual components, for example, differ from others by their texture, (black) pixel density, and size. Similar conclusions can be drawn for lines, circles and arrows. Consequently, reasonably discriminating features can be extracted from image data that will serve as input for the classification system.

We investigated two kinds of feature sets. On the one hand standard statistical features are calculated on image data. These measures are invariant in size and rotation. Roughly speaking they represent – to some extent – shape related properties of the analyzed connected components. Alternatively, intensities of gradient histograms (values ranging from 0 to 255, equally divided into 16 bins) of the connected components serve as features (gradient set).

The shape-set is based on the features proposed by Becker, the winner of the ICDAR 2005 Text Locating Competition [8]. They have been used successfully for natural scene text detection. In order to also cope with the detection and discrimination of graphical elements we extended the original set by certain additional statistical measures. In the reminder of this paper the first set of features is referred to as shape feature set or extended Becker set.

Original Becker features [8]:

Contrast.

$$F_{\text{contrast}} = \min\left(1.0, \frac{|\mu_{fg} - \mu_{bg}|}{20}\right)$$

Edge density.

$$I_{\text{density}}(x,y) = \sqrt{\left(\frac{1}{121} \cdot \sum_{p \in N_{11}(x,y)} I_{\Delta}(p)^2\right)}$$

$$avg\_density = \frac{\sum_{y=0}^{height-1} \sum_{x=0}^{width-1} I_{\text{density}}(x,y)}{width \cdot height}$$

$$F_{\text{edge\_density}} = \min\left(1.0, \frac{avg\_density}{10}\right)$$

Homogeneity.

$$F_{\text{homogeneity}} = 1.0 - \frac{\min\left(1.0, \frac{\sigma_{fg}}{180}\right) + \min\left(1.0, \frac{\sigma_{bg}}{300}\right)}{2}$$

Histogram overlap.

$$F_{\text{hist\_overlap}} = 1.0 - \frac{num_{overlap}}{num_{bg}}$$

Extended Becker set:

Becker features: (see above)

Canny edge intensity: The average of the intensity of an edge of the canny edge image I_{canny} within the area of a connected component's bounding box.

$$F_{\text{canny\_intensity}} = \frac{\sum_{x_{cc}}^{width_{cc}-x_{cc}} \sum_{y_{cc}}^{height_{cc}-y_{cc}} I_{\text{canny}}(x,y)}{weight_{cc} \cdot height_{cc}}$$

Number of foreground gray levels: The number of gray values in the foreground (i.e. on the connected component) of the bounding box of a graphical element.

$$F_{\text{fg\_gray\_values}} = \frac{\sum_{i;hist_{fg}[i]>0} 1}{\sum_{i;hist_{fg}[i]>0} 1 + \sum_{i;hist_{bg}[i]>0} 1}$$

Foreground mean gray level:

$$F_{\text{fg\_mean}} = \frac{\sum_{i} hist_{fg}[i] \cdot i}{\sum_{i;hist_{fg}[i]>0} 1}$$

Relative amount of gradient orientations: Using the histogram $hist_{angles}$ of the angles of the gradient image, the number of the angles appearing at least once is calculated.

$$F_{\text{gradient\_orientations}} = \frac{\sum_{i;hist_{angles}[i]>0} 1}{360}$$

Relative amount of foreground pixels: The number of pixels of the connected component divided by its area.

$$F_{\text{fg\_pixels}} = \frac{\sum_{i} hist_{fg}[i]}{width_{cc} \cdot height_{cc}}$$

Standard deviations : As the standard deviation is a measure of dispersion of data, in our case these features select irregular/textured connected components.

– Gray level intensity:

$$F_{\text{gray\_level\_deviation}} = \sigma(I(cc))$$

– Sobel gradient orientation:

$$F_{\text{gradient\_orientation\_deviation}} = \sigma(I_{\text{Sobel\_directions}}(cc))$$

– Sobel gradient magnitude:

$$F_{\text{gradient\_intensity\_deviation}} = \sigma(I_{\text{Sobel\_magnitudes}}(cc))$$

Using either the extended Becker set or the gradient set two alternative feature representations for connected components are extracted. In the first case input data is represented by a 12 dimension feature vector whereas in the latter the resulting feature space contains 16 elements.

The actual classification of the particular feature vectors is based on a Multi-Layer Perceptron (MLP). By means of cross-validation the network topology has been adjusted. We use one hidden layer with 15 or 20 neurons and the sigmoid function as activation function. Model training is based on standard backpropagation. The input and output of the network is defined by the number of input features calculated for each component (12 or 16 based on their nature) and the number of classes to be identified (4, i.e. arrows, circles/ellipses, lines, and text).

In order to deal with input data that does not belong to one of the known classes the following rejection strategy is considered. Let us denote by A_1 and A_2 the best two outputs of the classifier. The rejection function can then be defined as follows:

$$O(x) = \begin{cases} a_1 = c(A_1), & \text{if} |A_1 - A_2| \geq \epsilon \\ M + 1, \text{otherwise} \end{cases} \tag{1}$$

where $c()$ is the function, which gives the corresponding class for the outputs, a_1 is the top choice given by the classifier, M is the number of classes and $M + 1$ stays for the additional, rejection class. The ϵ is a parameter controlling the rejection rate ($0 \leq \epsilon \leq 1$).

3.3. Text Agglomeration

Once the classification of the different connected components is performed by the MLP, we can proceed to a higher level in the mindmap analysis. At this stage we step from a lower, connected component based level, to a higher one, which projects a sort of vague layout analysis as we can already distinguish between text and non-text elements (lines, circles, arrows). However, the primary goal is not to detect the layout but to merge different identified textual connected components into so-called “word structures”. This pseudo word level cannot really be equated with the physical word level as there is no information about what might be a word. This merging strategy is necessary for the further processing when a subsequently applied word recognition tool has to recognize the text.

Knowing that characters usually appear closer to each other than to other elements, by clustering they should group with their kind rather than with non-text elements. For that reason we discard all the items tagged by the classifier as being non-text and we perform a hierarchical clustering trying to merge the remaining items into words. Such an attempt can be observed in Fig. 3.

We have considered different distances in order to measure the similarity between two clusters. As we selected an agglomerative clustering strategy, we explored the suitability of the Euclidean distance between the physical center of the two connected components. A similar measure is computed for the gravity centers. Furthermore, the minimal distance between the boxes bounding graphical items is also considered. While these measures are easy to calculate their complexity is still high. Based on preliminary results we select the minimal distance to be considered for the further investigations.

A faster strategy is proposed by Yuan et al. [21], where the distance is based on the size of components to be merged as well as on the Euclidean distance between the components. In [5] the same idea was used successfully in a greedy clustering approach to separate text, drawings, charts, etc. Alternatively to the hierarchical clustering approach (see above) we explore the capabilities of greedy clustering thereby exploiting the following distance function:

$$f(s_1, s_2) = \sqrt{\frac{k s_1 s_2}{s_1 + s_2}}$$

where s_1, s_2 represent the sizes of the two connected components and k is a parameter controlling the level of the grouping. Analyzing the function f it becomes clear that it is rotation invariant, symmetric, and it does not respect just the distances but also the sizes of the components.

In order to use this measure, we calculate the distance between the components c_1 and c_2 having the size s_1 and s_2. If this distance is smaller than the value given by the function f then component c_1 and c_2 are merged and form a new cluster. This operation is iterated while all the unique components are tested.

4. Evaluation

In order to evaluate the effectiveness of the proposed system we pursued practical experiments on real whiteboard images. In the following we first give a description of the data set. Then classification results for connected components analysis are presented that illustrate the capabilities of the system to discriminate between text and non-text elements (circles, lines, arrows). Finally, results achieved by clustering the connected components are discussed.

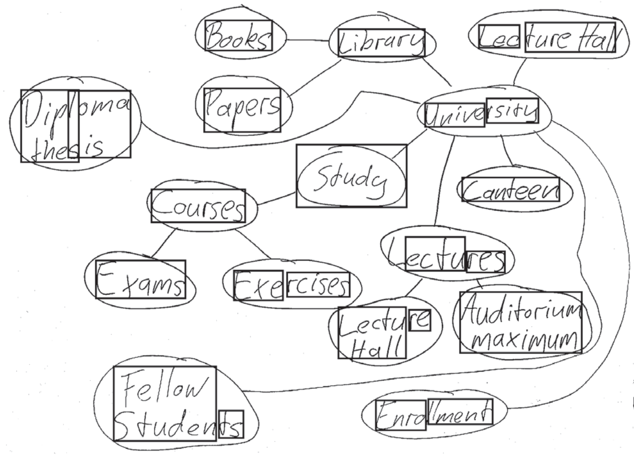


Figure 3. Results of text agglomeration (rectangles) on an exemplary mindmap image.

4.1. Data-set

Our whiteboard images based document set consists of 31 photos we took from mindmap-drawings on a whiteboard. Eleven different writers were asked to freely draw one mindmap for each of the topics ”holiday”, ”party” and ”study” (two writers sketched only two mindmaps). The writers were provided with a standard whiteboard marker set containing four different colors (black, blue, green, red) and a whiteboard eraser. Except for a basic set of words for each topic, which had to be used and an obligation to add at least three other words to the mindmap, there were no restrictions in creativity.

After a writer had finished his mindmap, a photo of the whiteboard was taken with a digital camera set to a resolution of 2,048x1,536 pixels. Reasoned by the image acquisition process, we can encounter in the picture items like the wall, other printed documents linked to the whiteboard, and frame parts of the board, which are not part of the document. These items are considered being noise elements (cf. e.g. [12] for a comparable argumentation).

In order to create the training and test we split the data randomly. The only constraint was to have one image sample from each of the writers. So we have a training set containing 20 images, while the test set consists of 11 images, one image from each particular writer.

4.2. Results

In the first experiment we evaluated the classification capabilities of the second stage of our segmentation approach, namely the analysis of connected components w.r.t. the discrimination between texts, circles, lines and arrows. We evaluated the two different feature sets described in Sec. 3.2. The overall classification accuracies (without rejection) for the extended Becker set and the gradient set are given in Tab. 1. It can be seen that both types of features are suitable for the classification. However, using the extended Becker features produces slightly better results. For the sake of clarity we limit the further presentation of results to this feature representation. The particular classification results of each writer can be seen in Tab. 2.

Features	Accuracy[%]
ext. Becker set(12)	95.7
gradient set(16)	93.0

Table 1. Results of connected components classification using different feature sets.

Writer	Accuracy[%]	Writer	Accuracy[%]
1	97.6	2	92.2
3	97.7	4	97.3
5	95.6	6	94.8
7	96.8	8	96.0
9	82.3	10	96.4
11	97.0		

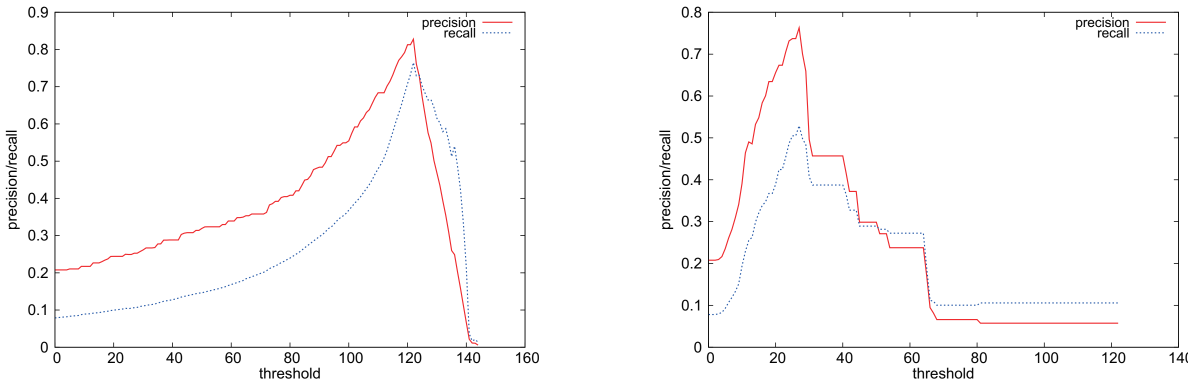
Table 2. Writer specific classification results for connected components analysis (extended Becker feature set).

While the text items are recognized with a high precision (99.4%) the arrows are often confused with lines. This confusion can be explained by the fact that just a few arrows are represented in our data set and there is not much difference between them. Similar confusions can be encountered for circles, related to text items where circles can be matched with textual elements like “o”, “D”, etc.

The rejection rates and the corresponding recognition accuracies for the different ϵ values (Eq. 1) are given in Tab. 3. Setting a stronger rejection criteria by increasing the parameter ϵ provides a higher accuracy, a decreasing misclassification rate and an increasing rejection percentage. However, the number of false positive rejections (given as absolute percentages in parentheses) also increases. Thus, care needs to be taken when adjusting the rejection threshold. For all further experiments reported in this paper we used small ϵ values, which practically implies no rejection.

For the evaluation of the quality of the agglomeration of textual elements we use the method introduced in the context of the ICDAR 2005 Text Locating Competition [8]. That way we produce comparable and comprehensible evaluation results.

The bounding boxes of the annotated ground truth T and



(a) Hierarchical agglomeration (b) Greedy clustering

Figure 4. Comparison of precision/recall values for one example document

ϵ	Accuracy[%]	Misclassified[%]	Rejection[%] (% FP)
0.1	96.2	3.8	1.1 (0.6)
0.3	97.1	2.9	3.5 (1.9)
0.5	97.9	2.1	6.3 (4.0)
0.7	98.7	1.3	10.9 (7.8)
0.9	99.5	0.5	22.8 (18.9)

Table 3. Dependency of classification accuracy on the choice of the rejection threshold

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:

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

The evaluation scheme is based on *precision* and *recall* known from the domain of Information Retrieval. 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:

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

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:

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

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:

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

We evaluated the output of the agglomeration using both schemes described above (cf. Fig. 4). In Fig. 4(a) we display a typical result of the hierarchical clustering, stating in this case the maxima for precision and recall at 83% and 72%, respectively. One can see that the other clustering method (cf. Fig. 4(b)) reaches almost the same precision value (76%) while the maximum recall is significantly lower (53%). Despite the worse overall results, this algorithm might be preferable in some cases as it obviously reaches the optimum a lot faster. These diagrams also illustrate the agglomeration process – starting with the initial component set and finishing with one huge cluster. As more and more components get agglomerated, the granularity of the clustering approaches its optimum. Further grouping leads to too large clusters and by that to worse precision and recall values.

5. Conclusion

In this paper we presented a segmentation approach for handwritten whiteboard notes that is based on a three-stage

(a) Touching items (b) False agglomeration for multi-line texts

Figure 5. Segmentation challenges

processing strategy. First we extract connected components, which are then classified w.r.t. belonging to known graphical elements or text. In order to obtain segmentation at word level in the final stage textual elements are merged by an automatic clustering procedure.

By means of an experimental evaluation we demonstrated the effectiveness of the proposed approach. We successfully extracted graphical and textual elements of handwritten mindmaps of real-world whiteboard images. Clustering of connected components identified as being text produced reasonable word level hypotheses. The latter can now serve as input for an actual handwriting recognition system.

Analyzing the segmentation results provided by the proposed system certain still remaining challenges can be identified (cf. Fig. 5). As illustrated in Fig. 5(a) touching connected components need to be separated properly. Furthermore, line separation is required for text portions before feeding them to an actual recognition system (cf. Fig. 5(b)).

In our future work we will address the aforementioned issues. Furthermore, we will consider to recognize the whole structure of the mindmap and the integration of the system with our handwriting recognizer [13].

References

[1] T. Buzan. *Business Mind Mapping*. Ueberreuter, 1999.

[2] H. Fujisawa. Forty years of research in character and document recognition - an industrial perspective. *Pattern Recognition*, 41(8):2435–2446, 2008.

[3] U. Garain and B. B. Chaudhuri. On machine understanding of online handwritten mathematical expressions. In *International Conference on Document Analysis and Recognition*, pages 349–353, 2003.

[4] B. Gatos, S. L. Mantzaris, S. J. Perantonis, and A. Tsigris. Automatic page analysis for the creation of a digital library from newspaper archives. *International Journal on Digital Libraries*, 3(1):77–84, 2000.

[5] W. Huang and C. L. Tan. A system for understanding imaged infographics and its applications. In *DocEng: ACM Symposium on Document Engineering*, pages 9–18, 2007.

[6] S. Klink, A. Dengel, and T. Kieninger. Document structure analysis based on layout and textual features. In *International Workshop on Document Analysis Systems*, pages 99–111, 2000.

[7] M. Liwicki and H. Bunke. Handwriting recognition of whiteboard notes. In *Conference of the International Graphonomics Society*, pages 118–122, 2005.

[8] S. M. Lucas. Text locating competition results. In *International Conference on Document Analysis and Recognition*, pages 80–85, 2005.

[9] A. M. Namboodiri and A. K. Jain. *Document Structure and Layout Analysis in Digital Document Processing: Major Directions and Recent Advances, B. B. Chaudhuri (ed.)*. Springer-Verlag, January 2007.

[10] W. Niblack. *An introduction to digital image processing*. Prentice Hall, 1986.

[11] A. Nomura, K. Michishita, S. Uchida, and M. Suzuki. Detection and segmentation of touching characters in mathematical expression. In *International Conference on Document Analysis and Recognition*, pages 126–130, 2003.

[12] D. M. Oliveira and R. D. Lins. Tableau - processing teaching-board images acquired with portable digital cameras. In *International Workshop on Camera-Based Document Analysis and Recognition*, pages 79–86, 2007.

[13] T. Plötz, C. Thurau, and G. A. Fink. Camera-based whiteboard reading: New approaches to a challenging task. In *International Conference on Frontiers in Handwriting Recognition*, pages 385–390, 2008.

[14] K. Roy, S. Vajda, A. Belaid, U. Pal, and B. B. Chaudhuri. A system for indian postal automation. In *International Conference on Document Analysis and Recognition*, pages 1060–1064, 2005.

[15] J. Sauvola, T. Seppanen, S. Haapakoski, and M. Pietikainen. Adaptive document binarization. In *International Conference on Document Analysis and Recognition*, volume 1, pages 147–152, 18–20 Aug. 1997.

[16] A. Sellen and R. Harper. *The Myth of the Paperless Office*. MIT Press, 2002.

[17] E. Tapia and R. Rojas. Recognition of on-line handwritten mathematical expressions in the e-chalk system - an extension. In *International Conference on Document Analysis and Recognition*, pages 1206–1210, 2005.

[18] K. Tombre and B. Lamiroy. Pattern recognition methods for querying and browsing technical documentation. In *Iberoamerican Congress on Pattern Recognition*, pages 504–518, 2008.

[19] J. van Beusekom, D. Keysers, F. Shafait, and T. M. Breuel. Example-based logical labeling of document title page images. In *International Conference on Document Analysis and Recognition*, volume 2, pages 919–923, sep 2007.

[20] D. Yoshida, S. Tsuruoka, H. Kawanaka, and T. Shinogi. Keywords recognition of handwritten character string on whiteboard using word dictionary for e-learning. In *International Conference on Hybrid Information Technology*, pages 140–145, 2006.

[21] B. Yuan and C. L. Tan. Multi-level component grouping algorithm and its applications. In *International Conference on Document Analysis and Recognition*, pages 1178–1181, 2005.