Semi-Automated Document Image Clustering and Retrieval

Markus Diem[a], Florian Kleber[a], Stefan Fiel[a] and Robert Sablatnig[a]

[a]Computer Vision Lab, Vienna University of Technology, Austria;

ABSTRACT

In this paper a semi-automated document image clustering and retrieval is presented to create links between different documents based on their content. Ideally the initial bundling of shuffled document images can be reproduced to explore large document databases. Structural and textural features, which describe the visual similarity, are extracted and used by experts (e.g. registrars) to interactively cluster the documents with a manually defined feature subset (e.g. checked paper, handwritten). The methods presented allow for the analysis of heterogeneous documents that contain printed and handwritten text and allow for a hierarchically clustering with different feature subsets in different layers.

Keywords: Document Clustering, Layout Analysis

1. INTRODUCTION

In 1989 shortly before the Fall of the Berlin Wall, Stasi officers tried to destroy secret records. About 600 million snippets of these records are preserved. They are now being processed in order to make them digitally accessible for registrars and in turn for the parties involved in the spying. The documents are either handwritten, copies of e.g. newspapers or carbon copies handwritten or printed text, typewritten or printed with dot matrix printers. After reassembling the snippets, tools are required that facilitate exploring the documents so as to establish links between the documents and ideally reproduce their initial bundling.

These tools should automatically create links between different documents based on their content. However, since handwriting of different authors is present in the documents and no layout constraints can be defined, OCR is not applicable at this processing stage. Thus, the links are established by means of a semi-automated document image clustering and retrieval. In order to achieve this, structural and textural features are extracted from the documents during a pre-processing stage. Then, the experts (e.g. registrars) can interactively cluster the reports with manually defined feature subsets. In addition, documents that have a similar (with respect to the features extracted) visual appearance can be retrieved.

This paper presents a set of features to capture visual similarity between documents and a semi-automated clustering and retrieval using this set of features. The features were on the one hand chosen so that each captures a specific visual feature and at the same time allow for the combination of these features. Hence, the features presented render use cases like *"Group all documents having similar supporting material color and writing"* possible. Having chosen a set of visual features, the user chooses the number of groups desired which allows for an interactive granularity change. In addition, the clustering can be performed hierarchically with different feature subsets in different layers. The features comprise layout characteristics like line spacing, character height, writing color, paper color, document type (form classification) and also a writer classification for handwritten text.

The paper is structured as follows. Subsequently in Section 2 the state-of-the-art in document image clustering and retrieval is presented. Then, in Section 3 the methodology – summarizing the features used and describing the clustering and retrieval – is presented. Section 4 comprises the evaluation of the features presented on the previously described dataset as well as competition results from publicly available datasets. Finally, a conclusion and future work is given in Section 5.

Further author information: (Send correspondence to Markus Diem)
Markus Diem: E-mail: diem@caa.tuwien.ac.at, Telephone: +43 (0) 1 588 01 183 55

2. LITERATURE REVIEW

In the literature, document clustering and retrieval are either developed for databases with a historical value[1] or databases that are not accessible for OCR due to their condition or writing.[2]

For OCRed documents, retrieval can be carried out using keyword searches. S. Karol and V. Mangat[3] propose document clustering using KPSO which is a Particle Swarm Optimization (PSO) initialized with a k-Means clustering. Since the content of the documents observed is already accessible, they cluster documents based on keywords that are extracted automatically. Similarly, M. Karthikeyan and P. Aruna propose a semi-supervised document clustering scheme based on k-Means clustering. In addition to the documents' contents they utilize Content Based Image Retrieval (CBIR) to support the document clustering if figures or images are present. The CBIR proposed uses major color sets and distribution block signatures for matching. X. Zhang et al.[4] cluster documents by means of a first-order Markov Random Field (MRF) that is labeled using relaxation labeling. The MRF links documents based on their content and links such as hyperlinks or citation links.

In contrast to these approaches, document image clustering and retrieval utilize image processing to compensate the lack of information that arises from the missing content which – for some document classes (e.g. historical documents, handwritten documents) – cannot be made accessible.

2.1 Document Retrieval

Given a query document, document image retrieval searches a database for relevant documents. Depending on the application, relevance is defined as textural similarity,[5] similarity in structure or layout[6,7] or documents with the same content[8,9] (e.g. same signature).

Back in 1997 J. Cullen et al.[5] proposed texture descriptors for document image retrieval. The texture descriptors are based on interest points that are detected using a Moravec Corner detector. Then, global features are computed based on the density of interest points, connected component sizes and the distribution of connected components. Finally relevant documents are retrieved by comparing the 80 dimensional global feature vectors using an Euclidean distance measure.

C. Shin and D. Doermann[6] propose a retrieval system based on layout similarity. For matching, they use spatial layout feature such as relative location and size. In addition, they incorporate information about the number and type of *"components"* and the column structure. The similarity measure is then calculated using region overlaps between the query and the retrieved image.

G. Zhu and D. Doermann[9] present document image retrieval based on signature matching. While they propose a sophisticated signature matching approach based on shape contexts and local neighborhood graphs, its use for general document retrieval is limited since signatures need to be present in both, the query and the retrieved document. Hence, this technique is especially beneficial if a document of a specific author needs to be retrieved.

A retrieval approach having a broader application is proposed by A. Gordo et al.[7] In contrast to the approaches previously mentioned, they focus on large-scale databases where the computation time of a single match becomes crucial. They propose runlength histograms that represent a document's structure at different scales. In order to reduce the size of their descriptors, they propose PCA embedding.

K. Takeda et al.[10,11] present a document retrieval that focuses on a single match between offline and online document data. They render real-time document retrieval possible by utilizing Locally Likely Arrangement Hashing (LLAH). Since the LLAH are computed locally, partly visible or occluded documents can be correctly retrieved. The similarity measure first matches local features using a hash table and then votes each match of a local feature.

2.2 Document Clustering

In contrast to document image retrieval, document clustering focuses on partitioning a query space into groups that have a predefined similarity. For historical manuscripts writer identification[12] and structural analysis[1] are computed to link the documents with each other.

M. Panagopoulos et al.[12] cluster 24 ancient Greek inscriptions with respect to the writing style. In order to determine similarities between the writing of different inscriptions, they extract ideal character prototypes using smoothed contours. Then, they form probability maps of the prototypes by means of their realizations. If one prototype letter is rejected by a statistical hypothesis test, the current observations are assumed to be written by different hands.

L. Wolf et al.[1] cluster historical manuscripts of the Cairo Genizah collection using manually assigned metadata (e.g. subject, script type) combined with visual features. The visual features are based on Bag-of-Features that are created using local descriptors.[13] Finally, they establish the similarity between documents by means of Support Vector Machines (SVM) that operate on the scores of multiple SVMs.

S. Chanda et al.[2] cluster torn documents to support forensic analysis of documents. Exclusively background information – namely the paper color and texture analysis – is incorporated in the similarity function. Finally they use Self-Organizing Maps (SOM) for partitioning the input space.

3. METHODOLOGY

As stated in the Introduction, we extract features that are interpretable for archivists. Therefore, experts may combine specific high level features to form clusters that fit their specific needs. The descriptors that represent these features are generally low dimensional so that their combination is not biased. First, information is extracted from the supporting material including its color and texture. Then the documents layout is extracted by means of a bottom-up approach that groups words to lines which are then grouped to paragraphs. Finally, the words are classified into no text or graphics, printed, and handwritten text. The descriptors are pre-computed for each document of a database and stored in an XML file which has typically a size of ≈ 1.27 KB.

3.1 Feature Extraction

Changes in the supporting material's color either arise from different manufacturing processes or aging effects. Hence, grouping documents having similar supporting material colors, allows for exploring similarly stored documents or documents that have a particular meaning (in modern document collections colored paper is used for e.g. title pages or separator sheets).

In order to extract a document's background color, the Luminance channel of the CIE L*a*b* color space is examined. A Gaussian Mixture Model (GMM) is fit to the luminance histogram to roughly partition the background pixels from the foreground pixels. Then, the mean color value of all background pixels is computed. However, faded-out ink impairs the estimation. That is why a weighted mean based on the pixels' gradient magnitudes is calculated which assigns a higher weight to homogeneous regions. A detailed description of the color extraction is given in Diem et al.[14]

In addition to the supporting material's color, its texture is analyzed. The texture analysis examines Fourier features[15] which are classified using multiple SVMs into lined, checked or blank paper. Since ruling is generally lighter than written text, a binarization is not suitable for segmenting ruling lines. Structures that are uniformly repeated (e.g. ruling) are local maxima in the Fourier space. Hence, we exploit the Fast Fourier Transform (FFT) for texture analysis of the supporting material. The texture features are normalized with respect to rotation by transforming the FFT space to a polar space, where each row represents one degree. Then, the row of the projection profile's p global maximum $max(p)$ and the corresponding perpendicular row $((max(p) + 90) \mod 180)$ are concatenated to represent supporting material that is lined, checked or has no texture. These features are finally classified using three one-against-all SVMs. Figure 1 illustrates the texture analysis worflow. First a 512×512 px image patch (b) with a maximal amount of background is extracted. This patch is then transformed to the Fourier space (c) which is further transformed to polar coordinates (d) to allow for an orientation invariant feature extraction.

Having analyzed the supporting material, the document's structure is analyzed. As previously mentioned, we employ a buttom-up approach which showed more robustness with respect to noise or poorly pre-processed images than top-down approaches. First, the characters are grouped to words in the binary image using Local Projection Profiles (LPPs). Issues arising from merged ascenders and descenders between text lines are resolved using a rough text line estimation which is based on a first derivative anisotropic Gaussian filtering. Then, locally

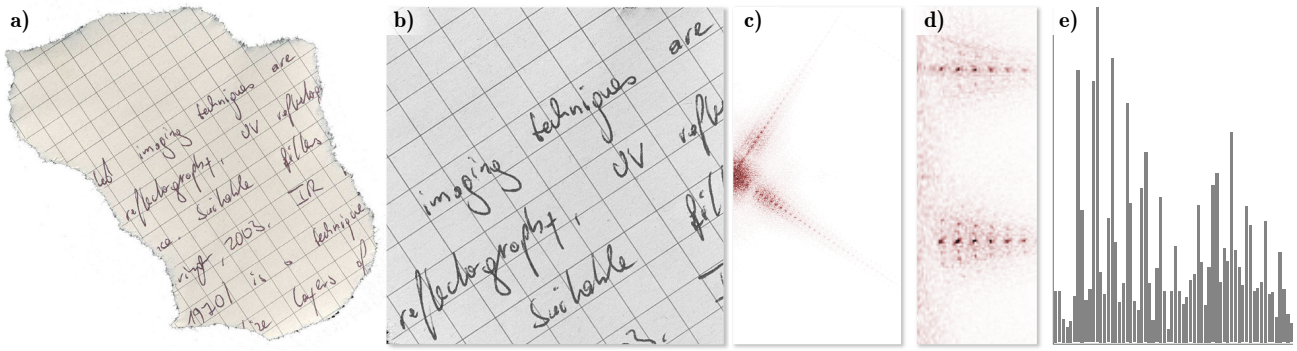

Figure 1. Torn document sample a) with the 512×512 *px* patch that contains most background b). Resulting Fourier space c) and polar transformed Fourier space d). Final feature vector e).

continuous local maxima are detected in the filtered image so as to split text lines that are merged. After these processing stages, the contour of words is known. In favor of processing speed and the complexity of subsequent algorithms, it is desirable to represent words rather by an enclosing rectangle than their contour. A rectangle that fits a words' x-height (body height) has advantages compared to approximations such as bounding boxes or the minimum area rectangles:

- the rectangle's orientation is similar to the word's local orientation
- the area covered minimizes the background since ascenders and descenders are neglected
- words with different local orientations or slant can be represented correctly

Due to these advantages, we introduced profile boxes (see Figure 2) that are computed by robustly fitting lines using the Welsch distance[16] to a word's upper and lower profile. Having detected both lines, the profile box is defined to have the mean angle of both lines, a height which is the mean distance between the lines and a width corresponding to the maximal length of both lines.

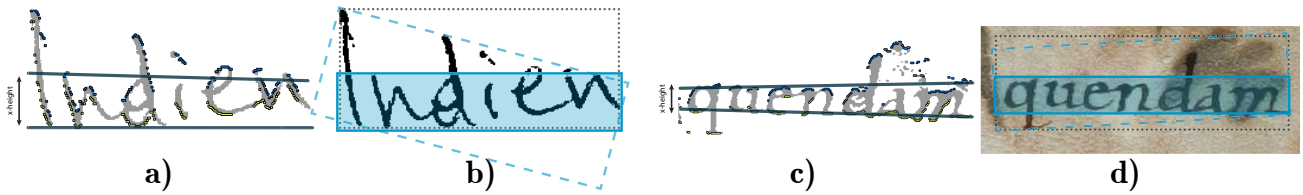

Figure 2. Upper and lower word profile a), c) with the corresponding upper and lower profile line. In b) and d) the resulting bounding box (dotted rectangle), minimum area rectangle (dashed line) and the proposed profile box (solid rectangle) are illustrated. Note that the profile box resembles the correct word orientation while having a minimal background which improves the descriptor extraction.

Considering a heterogeneous document collection, a binarization cannot correctly segment documents into *text/no text* since the grayscale information is not distinct enough. That is why we perform an additional feature extraction and classification that detects on the one hand graphical attributes, background noise (e.g. stains, printing artefacts) and on the other hand printed or handwritten text. The output of this classification are SVM weights that are used for the clustering stage. The feature extraction is performed using sliding windows along all profile boxes. Since the profile boxes adapt to the words local orientation and size, the feature extraction is robust with respect to scale and orientation. For every sliding window 64 dimensional Gradient Shape Features (GSF)[17] are extracted. The GSFs are adopted shape context feature which consider – similar to SIFT – the gradient magnitude rather than contour points. This allows for a robust feature extraction even if background

noise (e.g. carbon copies) is present. In order to compensate varying slant of handwritten words, the GSF's log-polar coordinate origin is chosen with respect to the word's dominant slant angle. The GSF descriptors are classified into three classes using SVMs with a one-against-one scheme. Voting the classification result of all descriptors within one word results in a final three dimensional word descriptor, where the dimensions represent either *noise/graphics, handwritten text, printed text.*

In order to extract features such as the text line frequency or mean word height, the words are merged to text lines and subsequently to paragraphs. During these stages the descriptors are voted so as to assign descriptors to text lines and paragraphs.

In addition to the structural and textural features, writer identification is computed for documents with more than 10% handwriting.[18] The writer identification is implemented by means of Bag-of-Words (BoW) using SIFT features. And a form analysis is carried out that allows to detect forms such as table of contents. A form is presented by a histogram of structural features of lines (solid and dotted) which have been trained offline for every form class. The structural (shape) features are based on the line information describing local line structures, e.g. line endings, crossings, boxes. The dominant line structures build a vocabulary for each form class. According to the vocabulary an occurence histogram of structures of form documents can be calculated for the classification and retrieval.

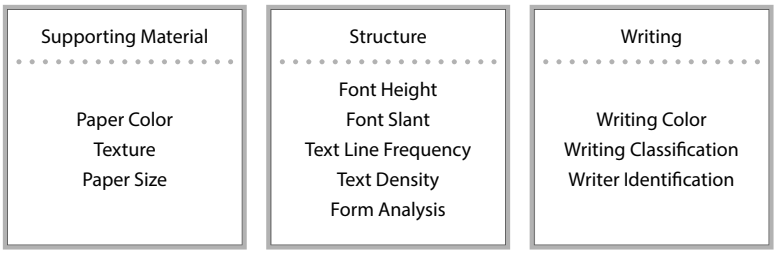

Figure 3. Clustering features grouped with respect to their origin.

Figure 3 shows the features that can be selected for clustering. The first column includes the supporting material features including the paper color, the three dimensional classification result of the texture analysis and the paper size. The paper size is especially useful to search for e.g. all DIN A4 pages within a dataset. The second column illustrates features derived from the document's structure. The font height is computed by the median x-height of all words. The slant is computed by the mean angle of the words' slants. It can be used to initially separate different handwritten pages. The text line frequency and text density allow for the separation of documents having varying layouts. The last feature derived from the document structure is the form analysis which is computed by extracting all lines of a document. The Writing feature analyze the structure and color of the writing. The writing color allows for grouping documents written with similar pens or ink color. Writing classification indicates the amount and probability of printed and handwritten text present in documents. Finally, the writer identification allows for grouping handwritten documents that are written by the same author. In total, a 16 dimensional feature vector can be generated if all features are selected. Note that form analysis and writer identification are the only features which cannot be combined, since they deploy high dimensional feature vectors which will therefore impair the clustering of all other features selected.

3.2 Document Clustering and Retrieval

The document image clustering aims at supporting experts in exploring large document databases that are digitally not made accessible. As previously mentioned, the features are assumed to be pre-computed and stored as XML files resulting in a low loading time.

The clustering is performed on all or any subset of the features listed in Section 3.1. The clustering is carried out using k-Means clustering which has some specific advantages compared to methods such as DBSCAN or SOMs. One of which is its fast processing time and that the user can choose the number of resulting groups. These can then be interactively adapted according to the user's needs. Figure 4 shows an example of clustering

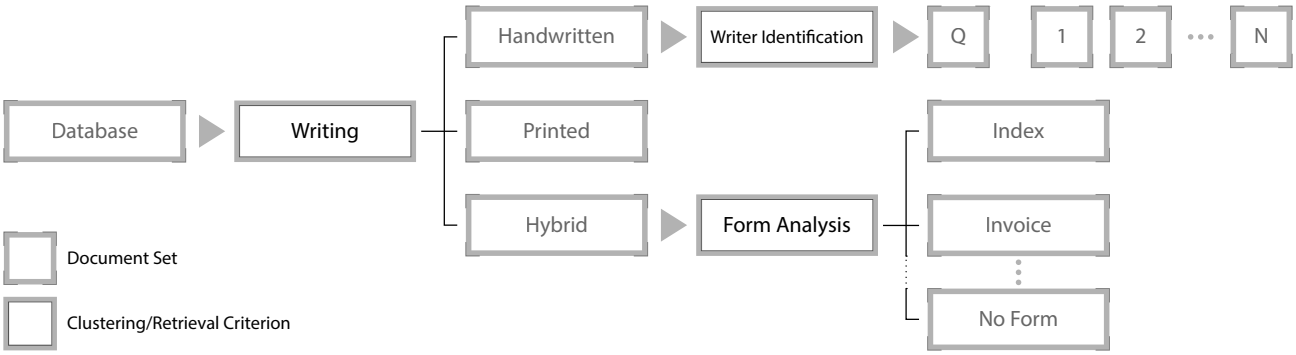

Figure 4. Illustration of hierarchical document clustering using different feature sets.

on subsets. First, the database is grouped into handwritten, printed text and documents that contain both. Then, a writer retrieval with a query image Q is performed on the subset containing solely handwritten text. The Hybrid subset that contains documents with printed and handwritten text is further subdivided using the form analysis.

To explore the document space, the user can choose which feature dimensions to combine when clustering. Additionally, the number of clusters can be chosen interactively. If a set of documents is formed (e.g. all handwritten documents), the user can again split this subset into smaller clusters having the same, different or additional feature properties. In addition, the retrieval can be performed either on the whole dataset or on individual clusters (e.g. writer retrieval may be carried out on handwritten documents only).

The image retrieval is performed by sorting all documents with respect to their distance from the query image. The distance measure is either an Euclidean distance or a cosine similarity for writer retrieval and form retrieval. If more than one query image is chosen, the retrieved documents are sorted according to their respective minimal distance to one of the query images. This allows for retrieving images with different features (based on the user selection) or to refine the retrieval result.

The clustering and retrieval application is designed to operate on current personal computers. Subsequently, some remarks on the applications ability to handle several images at the same time are given. The tests were performed using a notebook with an Intel T9900 @3.06GHz dual core with a total of 8GB RAM operating Windows 8. Loading the features of 1771 documents takes ≈ 4.97 *sec.* The application reserves ≈ 214 MB RAM for the same amount of documents with 160×160 *px* thumbnails that represent the documents. Clustering 1771 documents using 16 dimensional feature vectors takes 144 *msec* if 19 clusters are desired and 12 *msec* if the output is grouped into 2 clusters. The image retrieval takes 5 *msec* if one query image is provided and $344ms$ with 100 query images.

Figure 5 shows a user interface of the Clustering/Retrieval tool. Exemplarily, the documents are first grouped with respect to the paper size. Then 3 clusters are created containing handwritten text (blue), printed (yellow) and both text forms. In Cluster 3, a retrieval of white paper is performed.

4. RESULTS

In this chapter the features for clustering documents are evaluated rather than the clustering itself. The features are either evaluated on the Stasi dataset or on publicly available datasets. The former consists of documents which were written between 1950 and 1989. Hence, printed documents are either copied from e.g. newspapers, type written or printed with dot matrix printers.

4.1 Texture Analysis

The texture analysis was evaluated on 458 document snippets where it achieved a precision of 92.5%. Table 1 shows the confusion matrix of the classification results. Note that most confusions are between the *void* and *lined* class (0.117). This can be attributed to the fact that the dataset contains tables which are in these cases falsely classified as being *lined*.

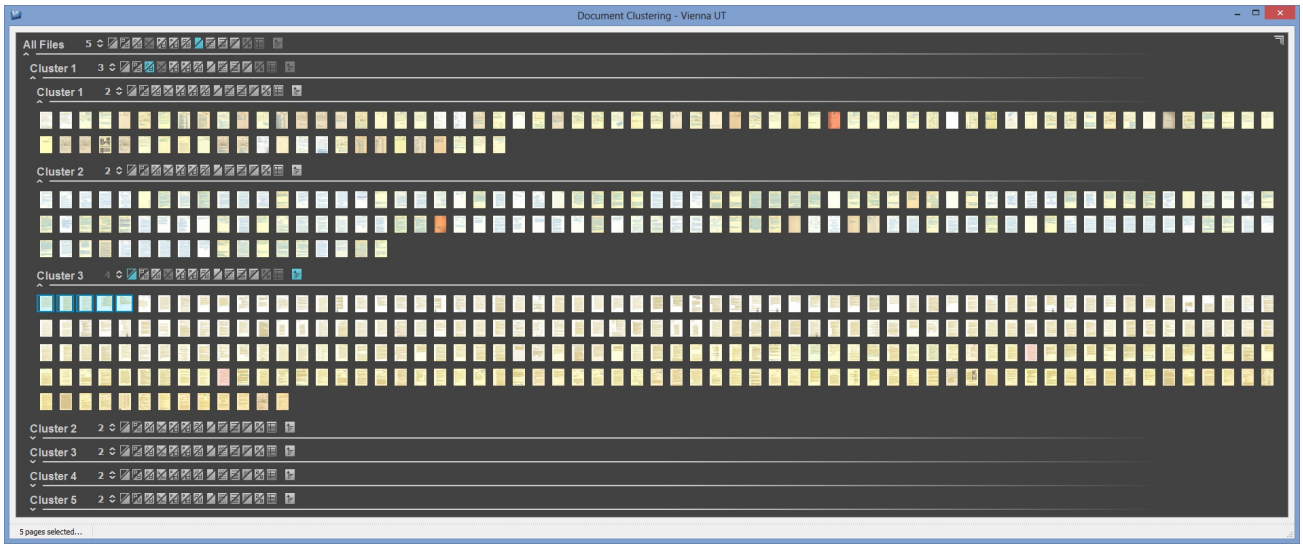

Figure 5. An example of the clustering user interface.

	predicted			
	void	lined	checked	#
void	**0.934**	0.029	0.047	314
lined	0.116	**0.884**	·	103
checked	0.049	·	**0.951**	41
	304	100	54	458

Table 1. The rows of the confusion matrix show the groundtruth labels, while the columns represent predicted labels (e.g. 11.6% of the lined paper is falsely classified as void).

4.2 Layout Analysis and Text Classification

The text classification and layout analysis engine was on the one hand evaluated on the Stasi dataset, on the other hand publicly available datasets were used for evaluation. The former consists of the previously described Stasi snippets with a total of 4821 words. In addition, the classification and layout analysis for printed documents was evaluated on the PRImA dataset which was used in the ICDAR 2009 Page Segmentation Competition.[19] This dataset consists of 55 document images including newspapers with complex layouts or scientific papers. Since no handwritten text is present in these images, the SVMs were trained with the class labels *graphics*, *printed* and *noise*. The third and fourth evaluation datasets are from the ICFHR 2010[20] and ICDAR 2009[21] Page Segmentation Contests. These datasets contain 200 and 100 handwritten document images respectively. Challenges in these datasets arise from changing text line angles, merged text lines and varying writing styles. Figure 6 shows example pages from the different datasets. First a snippet similar to those in the stasi dataset is presented in a). The red rectangles indicate falsely classified words whereas the green rectangles show correct classification results. In Figure 6 b) a sample page of the PRImA dataset is presented. Again green areas represent true positives while red areas indicate false positives. The transparent area shows true negatives. A handwritten sample page from the ICFHR 2010 Page Segmentation competition is presented in Figure 6 c) and d). The blue rectangles show the text lines that are located by merging the profile boxes. In d) the final line segmentation result is illustrated where different colors indicate different text lines.

The confusion matrix in Table 2 shows the text classification performance on real world data. On this dataset a precision of 92.4% is achieved. Note that *noise* has a lower classification performance compared to *print* and *manuscript*. This can be attributed to text bleed through which the system recognizes as either printed or handwritten text, but it is tagged as noise in the groundtruth.

Table 3 shows the F-scores of the ICDAR 2009 Page Segmentation competition where the proposed method is denoted by CVL. On this dataset an overall F-score of 94.47% was achieved. It can be seen that the two

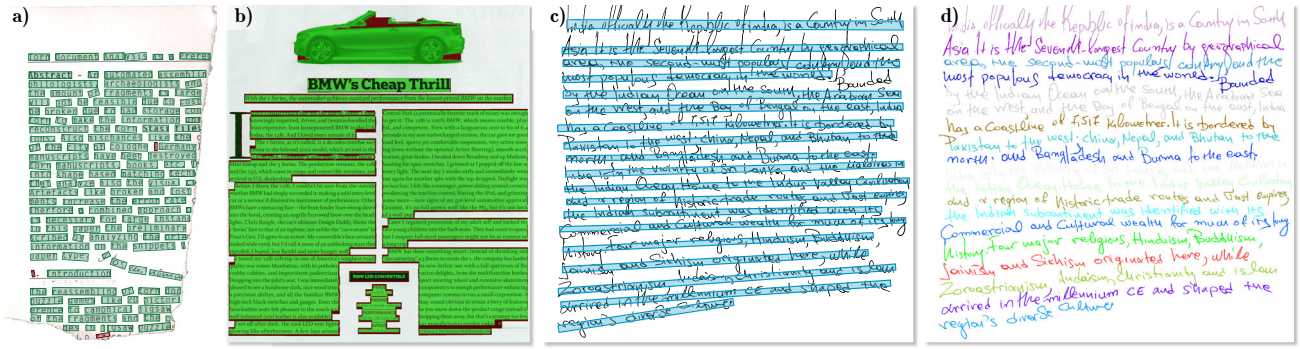

Figure 6. A sample image that is similar to the Stasi dataset a). A sample image taken from the PRImA dataset[19] b). And a sample image from the ICFHR 2010 Page Segmentation Competition[20] c) and d).

	predicted			
	noise	print	manuscript	#
noise	**0.625**	0.065	0.310	245
print	0.005	**0.945**	0.050	2180
manuscript	0.018	0.044	**0.938**	2034
	200	2166	2093	4459

Table 2. Text classification confusion matrix

	Non-text	Text	Overall
CVL	94.58	94.35	94.47
Fraunhofer	75.15	95.04	93.14
FineReader	71.75	93.09	91.90
Tesseract	74.23	92.50	91.04
DICE	66.22	92.21	90.09
REGIM-ENIS	67.13	91.73	87.82
OCRopus	51.08	84.18	78.35

Table 3. Page Segmentation Competition 2009.[19]

non-text classes *noise* and *graphics* allow for an accurate non-text estimation (94.58). Furthermore, Figure 6 b) shows that most errors rather result from an inaccurate border overlap between the methods output and the groundtruth than from missing or falsely classifed text boxes.

	DR	RA	FM
CUBS	97.54	97.72	97.63
NifiSoft	97.54	97.25	97.40
CVL	97.18	96.94	97.06
IRISA	96.87	96.45	96.66
ILSP-a	96.19	94.63	95.40
ILSP-b	95.70	94.20	94.95
TEI	95.09	94.62	94.86

Table 4. ICFHR 2010 Page Segmentation Contest.[20]

	DR	RA	FM
CUBS	99.55	99.50	99.53
ILSP-LWSeg-09	99.16	98.94	99.05
CVL	98.59	98.59	98.59
PAIS	98.49	98.56	98.52
CMM	98.54	98.29	98.42
CASIA-MSTSeg	95.86	95.51	95.68
AegeanUniv	77.59	77.21	77.40

Table 5. ICDAR 2009 Page Segmentation Contest.[21]

Table 4 and 5 give the results of the ICFHR 2010 and ICDAR 2009 Page Segmentation Contest respectively. In these evaluations the method's performance on segmenting handwritten text was evaluated. The performance metric is based on a MatchScore[20] that computes the maximum overlap of a text region with the ground truth region. If this score is above a given threshold T_α (which is 95% for text line detection), the text line is considered as correct (o2o). Based on this MatchScore, the Detection Rate (DR), the Recognition Accuracy (RA) and the Performance Metric (FM) are computed:

$$DR = \frac{o2o}{N}, \quad RA = \frac{o2o}{M}, \quad FM = \frac{2\ DR\ RA}{DR + RA} \tag{1}$$

where N is the number of ground truth text lines and M is the number of resulting elements. In other words, the DR can be considered as recall and the RA as precision. The proposed method achieves an F-Score of 97.06% and 98.59% on the ICFHR 2010 and the ICDAR 2009 dataset respectively.

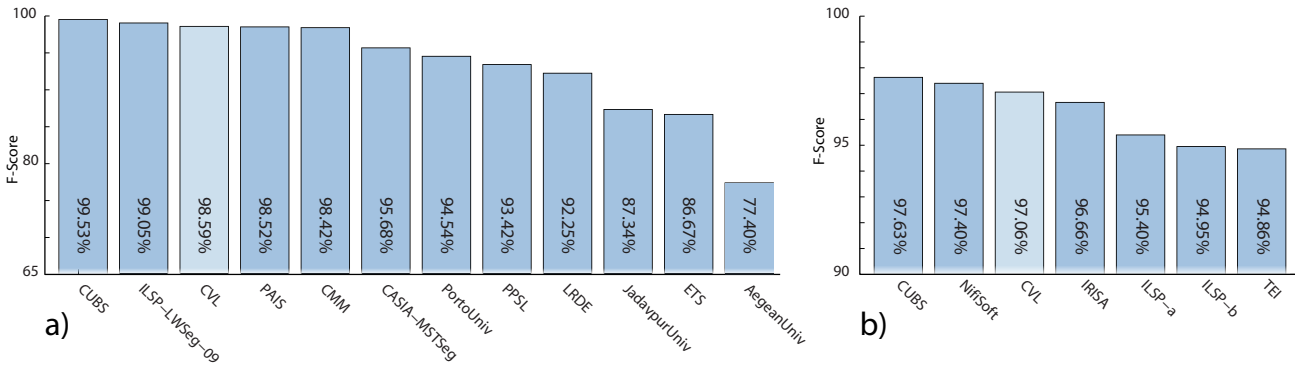

Figure 7. ICDAR 2009 page segmentation contest[21] in a) and ICFHR 2010 page segmentation contest in b).[20]

5. CONCLUSION

A semi-automated document image clustering and retrieval system was presented in this paper. The system is capable of clustering a huge number (≤ 5000) documents at the same time and thereby supports registrars in establishing links between documents if an unsorted dataset needs to be bundled.

It was shown in Section 4 that the visual features presented compete with other state-of-the-art methodologies. Further, the methods presented allow the analysis of heterogeneous documents that contain both, printed and handwritten text.

ACKNOWLEDGMENTS

The authors would like to thank the Fraunhofer-Institute for Production Systems and Design Technology (IPK), Berlin for supporting the work.

REFERENCES

1. L. Wolf, L. Litwak, N. Dershowitz, R. Shweka, and Y. Choueka, "Active clustering of document fragments using information derived from both images and catalogs," in *ICCV*, D. N. Metaxas, L. Quan, A. Sanfeliu, and L. J. V. Gool, eds., pp. 1661–1667, IEEE, 2011.
2. S. Chanda, K. Franke, and U. Pal, "Clustering document fragments using background color and texture information," in *DRR*, C. Viard-Gaudin and R. Zanibbi, eds., *SPIE Proceedings* **8297**, SPIE, 2012.
3. S. Karol and V. Mangat, "Evaluation of text document clustering approach based on particle swarm optimization," *Central Europ. J. Computer Science* **3**(2), pp. 69–90, 2013.
4. X. Zhang, X. Hu, T. Hu, E. K. Park, and X. Zhou, "Utilizing Different Link Types to Enhance Document Clustering Based on Markov Random Field Model With Relaxation Labeling," *IEEE Transactions on Systems, Man, and Cybernetics, Part A* **42**(5), pp. 1167–1182, 2012.
5. J. F. Cullen, J. J. Hull, and P. E. Hart, "Document image database retrieval and browsing using texture analysis," in *ICDAR*, pp. 718–721, IEEE Computer Society, 1997.
6. C. Shin and D. S. Doermann, "Document Image Retrieval Based on Layout Structural Similarity," in *IPCV*, H. R. Arabnia, ed., pp. 606–612, CSREA Press, 2006.
7. A. Gordo, F. Perronnin, and E. Valveny, "Large-scale document image retrieval and classification with runlength histograms and binary embeddings," *Pattern Recognition* **46**(7), pp. 1898–1905, 2013.
8. G. Zhu, Y. Zheng, and D. S. Doermann, "Signature-Based Document Image Retrieval," in *ECCV (3)*, D. A. Forsyth, P. H. S. Torr, and A. Zisserman, eds., *Lecture Notes in Computer Science* **5304**, pp. 752–765, Springer, 2008.
9. G. Zhu, Y. Zheng, D. S. Doermann, and S. Jaeger, "Signature Detection and Matching for Document Image Retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(11), pp. 2015–2031, 2009.
10. K. Takeda, K. Kise, and M. Iwamura, "Real-Time Document Image Retrieval for a 10 Million Pages Database with a Memory Efficient and Stability Improved LLAH," in *ICDAR*,[22] pp. 1054–1058.

11. K. Takeda, K. Kise, and M. Iwamura, "Real-Time Document Image Retrieval on a Smartphone," in Blumenstein *et al.*,[23] pp. 225–229.
12. M. Panagopoulos, C. Papaodysseus, P. Rousopoulos, D. Dafi, and S. Tracy, "Automatic Writer Identification of Ancient Greek Inscriptions," *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(8), pp. 1404–1414, 2009.
13. L. Wolf, R. Littman, N. Mayer, T. German, N. Dershowitz, R. Shweka, and Y. Choueka, "Identifying Join Candidates in the Cairo Genizah," *International Journal of Computer Vision* **94**(1), pp. 118–135, 2011.
14. M. Diem, F. Kleber, and R. Sablatnig, "Analysis of Document Snippets as a Basis for Reconstruction," in *Proceedings of the 10th International Symposium on Virtual Reality, Archaeology, and Cultural Heritage*, K. Debattista, C. Perlingieri, D. Pitzalis, and S. Spina, eds., pp. 101 – 108, 2009.
15. M. Diem, F. Kleber, and R. Sablatnig, "Document Analysis Applied to Fragments: Feature Set for the Reconstruction of Torn Documents," in *Proceedings of the 9th International Workshop on Document Analysis Systems*, D. Doermann, V. Govindaraju, D. Lopresti, and P. Natarajan, eds., pp. 393–400, (Boston, USA), June 2010.
16. R. E. Welsch and E. Kuh, "Linear Regression Diagnostics," Tech. Rep. 923-77, Massachusetts Institute of Technology, April 1977.
17. M. Diem, F. Kleber, and R. Sablatnig, "Text Classification and Document Layout Analysis of Paper Fragments," in *International Conference on Document Analysis and Recognition*,[22] pp. 854–858.
18. S. Fiel and R. Sablatnig, "Writer Retrieval and Writer Identification Using Local Features," in Blumenstein *et al.*,[23] pp. 145–149.
19. A. Antonacopoulos, S. Pletschacher, D. Bridson, and C. Papadopoulos, "ICDAR 2009 Page Segmentation Competition," in *ICDAR*, pp. 1370 –1374, jul. 2009.
20. B. Gatos, N. Stamatopoulos, and G. Louloudis, "ICFHR 2010 Handwriting Segmentation Contest," in *ICFHR*, pp. 737–742, 2010.
21. B. Gatos, N. Stamatopoulos, and G. Louloudis, "ICDAR 2009 Handwriting Segmentation Contest," in *ICDAR*, pp. 1393–1397, 2009.
22. *2011 International Conference on Document Analysis and Recognition, ICDAR 2011, Beijing, China, September 18-21, 2011*, IEEE, 2011.
23. M. Blumenstein, U. Pal, and S. Uchida, eds., *10th IAPR International Workshop on Document Analysis Systems, DAS 2012, Gold Coast, Queenslands, Australia, March 27-29, 2012*, IEEE, 2012.