A FRAMEWORK FOR DOCUMENT PRE-PROCESSING IN FORENSIC HANDWRITING ANALYSIS

Franke, K.; Köppen, M.

Published in:
EPRINTS-BOOK-TITLE

A framework for document pre-processing in forensic handwriting analysis

Katrin Franke, Mario Köppen
Fraunhofer Institute for Production Systems and Design Technology IPK,
Pascalstr. 8-9, D-10587 Berlin, Germany,
Fon: +49 30 39006194, Fax: +49 30 3917517,
E-mail: {katrin.franke, mario.koeppen}@ipk.fhg.de

We propose an open layered framework, which might be adapted to fulfill sophisticated demands in forensic handwriting analysis. Due to the contradicting requirements of processing a huge amount of different document types as well as providing high quality processed images of singular document classes, neither a standardized queue of processing stages and fixed parameter sets nor fixed image operations are qualified for such a framework concept. The open layered framework, proposed in this paper, provides adaptation abilities at the parameter level, the operator level and the algorithm level. Moreover, an embedded module that uses genetic programming might generate specific filters for background removal on the fly. In the following the layered framework will be presented, aspects of the implementation and results of its application will be given.

1 Introduction

Forensic handwriting examination, usually performed on request of the justice or private persons, deals with the analysis and the evaluation of handwriting to detect frauds. According to the lack of objective measurements and reproducible decisions, traditional methods, like visual inspection and expert rating, were tried to replace by computerized semi-automatic and interactive systems. Two systems operating in forensic labs for such a purpose are the FISH-system[1] and the SCRIPT-system[2].

Computer supported investigation of handwriting, using methods of digital image processing and pattern recognition, requires sophisticated pre-processing functions to remove background textures of documents and other imprints, which are not subject of the examination. Unfortunately, at the beginning of our project in 1997, there wasn't a suitable solution available, which was able to process huge amounts of different document types as they occur in daily forensic casework and which would provide high quality handwriting images as they are needed for further examinations[3]. Furthermore, major drawbacks were caused by only partially eliminated noise signals, cases where a correct separation of handwriting from textured background was impossible and cases where parts of the handwriting were lost, in particular those with a low contrast

In: L.R.B. Schomaker and L.G. Vuurpijl (Eds.), Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition, September 11-13 2000, Amsterdam, ISBN 90-76942-01-3, Nijmegen: International Unipen Foundation, pp 73-82

to the background.

To overcome the mentioned disadvantages a research project was initiated by the Bundeskriminalamt (the German central police bureau). The aim was to design and to realize a new framework for the elimination of noise signals from digitized handwritten documents that are subjects of forensic examination. Moreover, the new framework should be able to be integrated into an existing system environment like the FISCH-system[1] and it should be able to use the framework as a stand-alone application as well. The practical relevance of the project might be pointed out with some facts. The FISH-system[1] was conceptualized to handle 10.000 investigation cases per year. This means in the worst case there would be 10.000 different document types. There are no restrictions to the types of documents. The only thing what they have in common is handwriting on them. 77.000 documents were stored until 1995 in the FISCH-system[1] and this number increases every day. The scientific challenge is caused by this huge amount of different documents, being assumed as infinite. Moreover, at the same time high quality images of handwriting must be provided even if there are low contrast strokes and/or textured background (figure 1).

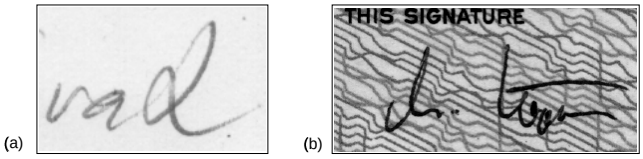

Figure 1: Parts of handwriting: (a) including low contrast strokes; (b) written on textured document background

From the authors point of view the realization of fully document-independent processing operators for handwriting segmentation will stay a scientific challenge for a while. A standardized queue of processing stages can not be used as a new framework concept. Also, fixed parameter sets are not useful. Even the number of different image operations should be extendable, because there are many approaches promising good processing results[4 5 6]. The best solution seems to be an open and partially interactive assistance system, including a functional kernel that provides basic image operations for the most frequent document types as well as opportunities to adapt/extend the functionality by user interactions.

From that required modularity, an open layered architecture (figure 2) was derived by the authors. Each layer stands for an abstraction level of procedures working on the documents. The layers support the understanding

of how the framework operates and how user interaction might adapt/extend the functionality to specific demands.

In the following section the layer model of the framework is motivated and explained as a whole. Then, section 3 focuses on single layers and section 4 provides some facts about the implementation. Results of the framework application were stated in section 5, followed by conclusions and further studies in section 6.

2 Layer model of the framework for document pre-processing

The proposed framework, whose block diagram is shown in figure 2, is made of the *IFH* (image and file handling), the *PS* (parameter specification), the *OA* (operator adaptation) and the *AS* (algorithm specification) layers. With respect to this huge amount of different documents the *LR* (layout recognition) layer is not working autonomously. Here, the user's intelligence is still required.

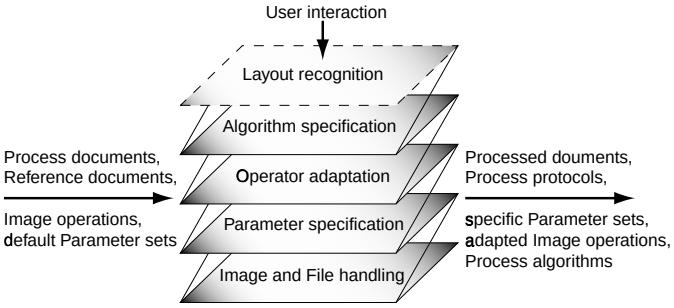

Figure 2: Layer model of the framework for forensic document processing

The on-line phase of the document processing works as follows: From a collection of pre-defined pattern documents, a human user selects the suitable one, which includes all that information needed to process the document(s) correctly. The information stored in the pattern document includes the structural and numerical parameters, the involved image operators as well as the processing flow. So, the on-line phase operates autonomously. Documents with their corresponding pattern documents might be processed on a remote machine or as batch job over the night.

In the off-line phase the mentioned pattern documents might be generated as well as small numbers of documents can be processed interactively, whereby a protocol might be obtained in the same style as a usual pattern document.

Moreover, in the off-line phase the user can also generate new image operators, which are adapted to special demands that are not predictable during the framework development. A detailed description of all the layers, supporting user interaction in the off-line phase will follow below. Before this, the approach will be motivated.

Launching the reseach project, 212 different document types were provided by the Bundeskriminalamt. The document types were selected from various backgrounds and cover memos, diverse bank formularies, passports, contracts, delivery notes, invoices, applications for a work permit, hotel registration etc. The majority of documents was sized between DIN A6 and DIN A4. The documents were sorted into three groups. Types of documents being examined very often represent group One. For this group it should be easy to get an empty reference document. The second document group includes such documents, which occur more or less often. In some cases it might be difficult to obtain an empty reference document. However, it would be nice to create a pattern document, because the document type has to be examined frequently. Then, a pre-created pattern document would help to save time and high quality images could still be provided. The third document group covers all that document types appearing only few times and/or there is no way to get an empty reference. From these three document groups the required image operations were derived. The image operators and finally the types of document processing differ in their consideration of a-priori knowledge (e.g. empty reference, layout information and/or processing parameters). In the following the types of document processing are listed in an increasing order by the strength of their connectivity to a reference:

- document-independent (without layout knowledge)
- document-specific (by using an adapted set of parameters)
- document-dependent (by using an empty reference template)

The required processing quality of the handwritten documents and the flexibility of the whole framework are opposing requirements (see figure 3). Therefore, the image operators to be included into the framework have to be selected carefully.

The image operators chosen by the authors cover local and global image filters, textural and structural as well as syntactical and layout driven approaches. More specialized, document-independent operators were provided for background removal of homogeneous and textured backgrounds (samples can be seen in figure 1) and for foreground removal of lines, machine-prints and noise. The document-specific processing is realized by using special parameter sets as an additional input for the before mentioned document-independent

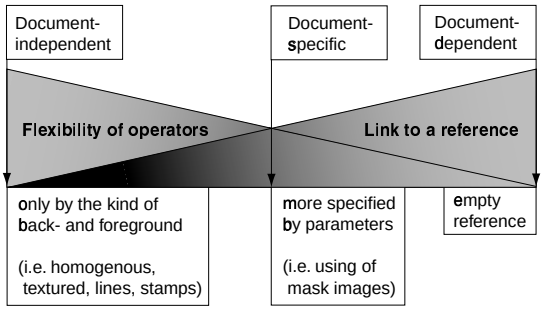

Figure 3: Types of document processing

operators. Moreover, an embedded module (see figure 4) allows the automated generation of document-specific background filters. The document-dependent processing is realized by using the approach of morphological addition[7 8].

In the literature diverse approaches for the pre-processing of handwritten documents were published[4 5 6]. Especially, color image pre-processing promises sophisticated results. The presented open layered architecture of the proposed framework is able to consider them. Within each layer of the framework new approaches might be included and the users of the framework might interact with them.

3 Layer descriptions

The IFH Layer for image and file handling: Dealing with digitized documents requires managing them. Users want to scan, load, view, browse, save and process document images. A huge database is not part of the presented framework. Depending on the application it is readily available, like in the case of the FISH-system, or it might be extended. The mentioned pattern documents as well as the protocol files were stored as **h**yper**t**ext **m**arkup **l**anguage-(html)-files. Of course, these html-files are generated automatically. The advantage of using them is that fixed relation of document images to all other parameters and the files can be distributed and displayed easily.

The PS Layer for parameter specification: Assigning different image operators to various regions of interest within a document is a common way. Within the PS-layer these structural parameters, also called mask images, can be defined interactively. Then, the specific parameter sets, the operators and the algorithms have to be assigned to each region. There are also some adjustable numerical parameters, e.g. the maximal noise ratio.

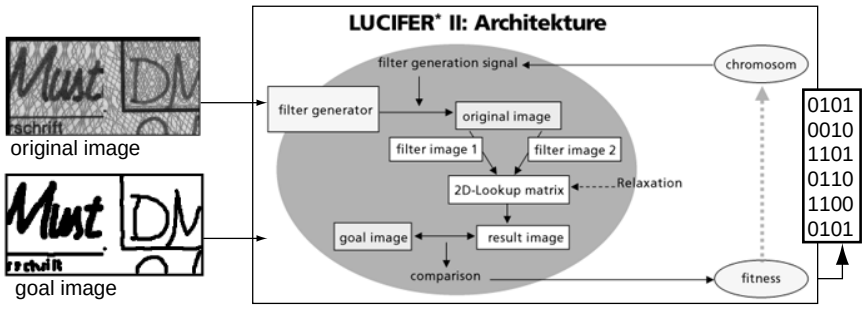

Figure 4: Embedded LUCIFER-module for background filter generation

The OA Layer for operation adaptation: To extent the functional kernel, a module for the automated generation of image processing operators is embedded within the framework. This LUCIFER-module[9 7] (see figure 4) uses genetic programming to design texture filters, which can be integrated in the document processing for background filtering. The user has only to provide a so-called goal image as well as a small part of the original image, from which the handwriting can not be extracted using the other background filters. The goal image is human made by tracing the foreground information, in particular the handwriting. In this way the goal image might verify the filters generated by the LUCIFER-module. A detailed description can not be given here. Further information might be found in[9 7].

The AS Layer for algorithm specification: The open architecture of the framework allows the use of various approaches to document processing. Unfortunately, there is no really intelligent layout recognition yet, which would be able to process this huge amount of different documents. Therefore, a human user has to do this job once in advance. The image operators, implemented in the framework, are chosen in such a way that users are not overwhelmed, hopefully. They can switch on/off single operations like:

- Homogenous background + grayvalue output image
- Textured background + lines + noise + binary output image
- Generated filter XYZ + machine-print + noise + binary output image
- Empty reference dropout (reference ABC) + noise + colorized output image

Even if the framework will be extended, this procedure will hold.

4 Implementation

The functional framework-kernel, which includes all modules for digital image processing was implemented as a library written in ANSI C. Up to now it runs under UNIX (Solaris 2.5.1 and SUN OS 4.3), WIN NT 4.0 and OS/2 Warp 4.0 operating system. Due to a platform and operating system independent software design it also can be used on a PowerMacintosh under MacOS 8.0. The major part of the graphical user interface was implemented in Smalltalk using ParcPlace Visual Works 2.5 with C-Connect running on various platforms, too. This interface covers the management functions for process images, mask images and html-protocol-files, as well as the kernel-function calls and the inter-application communication.

Due to the software design with its strict separation of functional kernel and user interface further applications were implemented, adapted to special demands in daily forensic casework. The first one allows to process huge amounts of documents and is realized as a simple console application running on a remote machine. For small forensic labs with limited resources and small amounts of documents to be processed an Adobe(TM) Photoshop(TM) Plug-In was implemented. With respect to the programmer interface of Photoshop(TM) the Plug-In covers only the document-independent processing functions for homogenous and textured backgrounds as well as for line, imprint and noise removal. Beside complete applications, the functional kernel was distributed as a software library with programmer interface and is now operating in automated check processing systems.

5 Results

A flatbed scanner supporting 256 grayvalues and 300 dpi resolution is used to digitize the handwritten documents. Grayvalue images were chosen with respect to the expected high quality images and the limited hardware resources. Therefore, up to now, the proposed framework includes only grayvalue image processing operators.

To overcome the mentioned problems of low contrast strokes a new approach, which was presented elsewhere[10], was implemented by using methods of mathematical morphology. The main processing stages are contrast enhancement, local adaptive binarization and reconstruction. To obtain high quality results, the first processing stage is most important. What is lost, will never be restored. The latest results of studying the enhancement of low contrast strokes are given in figure 5.

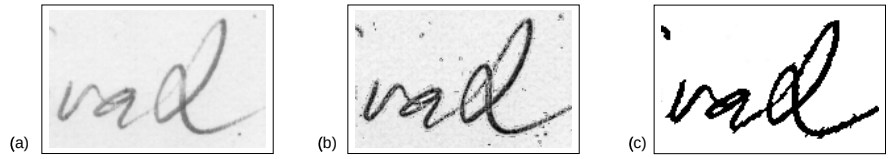

Figure 5: Processing of low contrast strokes: (a) original handwriting image; (b) contrast enhanced image ; (c) binarized and reconstructed image

The processing of textured documents was studied intensively in the past[7] [10]. Textured background removal on grayvalue images is a challenge. Handwriting might not be separated by global grayvalues intensities or empty reference subtractions[7]. Only, the consideration of local structures and local intensities by applying adapted morphological operators provided acceptable results (compare figure 6). The first implementation of this filter was done by the authors and proven using the LUCIFER-module[9 7]. For any further requests, like background removal of a new passport, applying the LUCIFER-module seems to be more convenient.

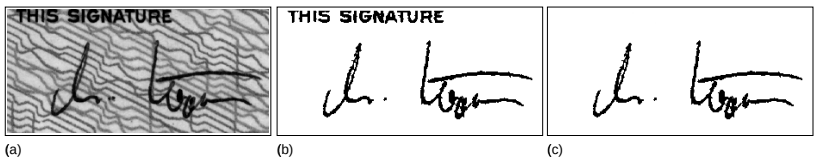

Figure 6: Processing of textures background: (a) original handwriting image; (b) binarized image after background removal; (c) handwriting after line removal

The total processing time for a document differs due to varying conditions. Average processing time for some samples processed on a Pentium II with 266 MHz and 196 MByte RAM are listed in table 1. Note that currently the homogeneous background removal is more sophisticated and able to keep low contrast strokes whereas texture background removal allows the extraction of handwriting from textured background with comparable lesser quality.

6 Conclusions and future work

A framework was proposed and implemented for the automated pre-processing of documents for forensic handwriting analysis. Fulfilling the stated requirements this framework can be used in the daily casework of forensic experts.

document	size	time
Homogenous background		
A4 page	2478 × 3469 Pixel	40.959 sec
bank cheque	1771 × 1006 Pixel	8.592 sec
snippet	672 × 227 Pixel	0.640 sec
Texture background		
A4 page	2478 × 3469 Pixel	17.085 sec
bank cheque	1771 × 1006 Pixel	3.435 sec
snippet	672 × 227 Pixel	0.220 sec

Table 1: Processing times for background removal of various documents with 256 graylevels.

The framework covers functions for digitalization of documents, their pre-processing, in particular the extraction and qualitative improvement of handwriting, and the archiving of processing protocols and processing parameters. The open architecture of the functional kernel supports the adaptation to further demands.

The document pre-processing itself follows a new concept that considers different kinds of bindings of a-priori knowledge to processing documents. Within the concept it is distinguished between document-independent processing without considering a-priori knowledge, document-specific processing with a binding of parameter sets to a document and document-dependent processing with a strict binding of a reference to a document. To provide a wide variability of processing filters the basic functional kernel might be extended by user generated filters using the LUCIFER-framework[97] working as an embedded module.

The current restriction to graylevel image processing caused by limited hardware resources seems to be irrelevant in the future. To improve the processing quality, color image processing[4] has to be considered in framework updates and extensions.

Acknowledgments

The authors wish to thank Yu-Nong Zhang, Xiufen Liu and Wensheng Ye for their contributions in coding and testing. Also, thanks to Gerhard Grube and all other forensic handwriting examiners, who were involved in verifying the framework. This work was sponsored in part by the Bundeskriminalamt, section handwriting examination (GZ.:IX/B-2310297-Ge).

References

1. Manfred Philipp. Fakten zu FISH, Das Forensische Informations-System Handschriften des Bundeskriminalamtes - Eine Analyse nach über 5 Jahren Wirkbetrieb. Technical report, Kriminaltechnisches Institut 53, Bundeskriminalamt, Thaerstraße 11, 65173 Wiesbaden, 1996.
2. Wim C. de Jong, Leny N. Kroon van der Kooij, and Dick Ph. Schmidt. Computer aided analysis of handwriting, the NIFO-TNO approach. In *4th European handwriting conference for police and government handwriting experts*, 1994.
3. Yan Solihin and C.G. Leedham. Preparing handwriting images for forensic examination. In *Proceedings of the Eight Biennial Conference of the International Graphonomic Society*, pages 125–126, 1997.
4. C. Cracknell and A.C. Downtown. A color classification to form dropout. In *Proceedings 6th International Workshop on Frontiers in Handwriting Recognition (IWFHR)*, Tajon, Korea, 1998.
5. K. Yoshikawa, Y. Adachi, and M. Yoshimura. Extracting the signatures from travelers' checks. In *Proc. 6th International Workshop on Frontiers in Handwriting Recognition (IWFHR)*, Tajon, Korea, 1998.
6. X. Ye, M. Cheriet, C.Y. Suen, and K. Liu. Extraction of bankcheck items by mathematical morphology. *International journal on document analysis and recognition*, 2(2/3):53–66, 1999.
7. Katrin Franke and Mario Köppen. Towards an universal approach to background removal in images of bankchecks. In *Proceedings 6th International Workshop on Frontiers in Handwriting Recognition (IWFHR)*, Tajon, Korea, 1998.
8. M. Okada and M. Shirdhar. Extraction of user entered components from a personal bankcheck using morphological substraction. *International journal of pattern recognition and artificial intelligence*, 11(4):549–570, 1997.
9. Mario Köppen, Martin Teunis, and Bertram Nickolay. A framework for the evolutionary generation of 2d-lookup based texture filters. In Takeshi Yamakawa and Gen Matsumoto, editors, *Methodologies for the Conception: Design and Application of Soft Computing, Proceedings of the 5th International Conference on Soft Computing and Information / Intelligent Systems (IIZUKA'98)*, pages 965–970, 1998.
10. Katrin Franke and Gerhard Grube. The automatic extraction of pseudo-dynamic information from static images of handwriting based on marked grayvalue segmentation. *Journal of forensic document examination*, 11(3):17–38, 1998.