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

A computer-based system to support forensic studies on handwritten documents

Katrin Franke, Mario Köppen

Fraunhofer Institute for Production Systems and Design Technology, Department Pattern Recognition, Pascalstr. 8-9, 10587 Berlin, Germany
e-mail: {katrin.franke|mario.koeppen}@ipk.fhg.de

Received: December 4, 2000 / Revised version: date

Abstract. Computer-based forensic handwriting analysis requires sophisticated methods for the pre-processing of digitized paper documents, in order to provide high-quality digitized handwriting, which represents the original handwritten product as good as possible. Due to the requirement of processing a huge amount of different document types neither a standardized queue of processing stages, fixed parameter sets nor fixed image operations are qualified for such pre-processing methods. So, we present an open layered framework that covers adaptation abilities at the parameter, operator and algorithm level. Moreover, an embedded module, which uses genetic programming, might generate specific filters for background removal on the fly. The framework is realized as an assistance system for forensic handwriting experts and it is in use by the Bundeskriminalamt, the federal police bureau in Germany, for one and a half years. In the following the layered framework will be presented, fundamental document-independent filters for textured, homogenous background removal and for foreground removal will be described as well as aspects of the implementation. Also, results of the framework-application will be given.

a)

b)

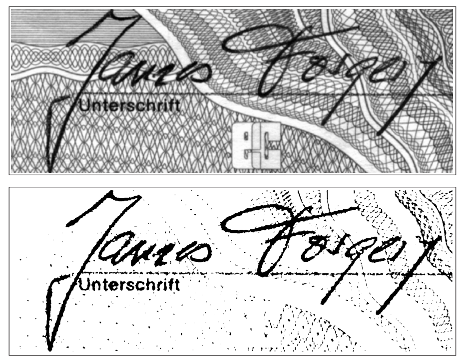

Fig. 1. Poor separation of handwriting from textured background: (a) original signature image, (b) signature image after procesing

1 Introduction

Handwriting examination or identification is frequently used in crime investigation, the prosecution and sentencing of criminal offenders. Important fields of interest are for example financial fraud, extortion, threat, terrorism, drugs related crime, pornography and racism. Aims, methods and techniques of forensic handwriting examination differ fundamentally from those of graphology. A forensic handwriting expert compares handwriting on the basis of well-defined sets of characteristics and does not make any relation between handwriting characteristics and personality traits.

Forensic handwriting examination has become an established part of forensic science. As other related fields of forensic science, like fingerprint identification, examination of toolmarks and bullets, handwriting examination is primarily based upon the knowledge and experiences of the forensic expert. Although much effort has been put into the training of the examiners, the methodology of forensic handwriting analysis is being criticized, in particular the lack of proof of validation. According to the non-objective measurements and non-reproducible decisions, traditional methods, like visual inspection and expert rating, were tried to be supported by computerized semi-automatic and interactive systems. Two systems operating in forensic labs for such a purpose are the FISH-system [21] and the SCRIPT-system [4]. Although there is some promising research in this area [7], [8], [24], it is rather rarely applied in the forensic labs.

Computer-supported analysis of handwriting, using methods of digital image processing and pattern recognition, requires sophisticated pre-processing methods to extract the handwriting specimens from documents, in particular to remove the document background as well as other imprints being not subject of the handwriting examination. Unfortunately, the systems that were used in forensic labs [21] [4] could not fulfill this demand properly (figure 1 and figure 2). On request of the handwriting examination group of the Bundeskriminalamt (the German police bureau) an open layered framework was developed to process handwritten documents.

The key idea of the model is to provide distinguished levels of user interaction during the configuration of the system for document processing, in particular from a simple parameter adjustment up to an initialization for an automated filter generation. In contrary to known approaches that scope on parallel processing [1] the layers of the proposed model

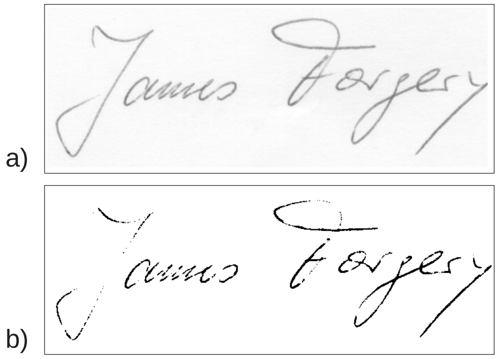

Fig. 2. Poor threshold binarization: (a) original signature image, (b) signature image after processing

represent abstraction levels of potential user interaction. Of course, the overall processing of a certain document is carried out automatically.

In the following section the layered model of the proposed framework is motivated and explained as a whole. Then, section 3 focuses on single methods of handwriting pre-processing that are applied within the framework. Such methods that are not published yet will be presented into detail. Section 4 provides some facts about the implementation. Results of the framework's application are stated in section 5, followed by conclusions and further studies in section 6.

2 System overview

Within this section the developed framework will be presented from various perspectives. At first a brief description of the major elements of the user interface will be given. So, the reader gets familiar with the available tools. The second point of view comprises the on- and off-line phase of the framework application, whereby the third view of the framework focuses on its abstract architecture to explain user interaction abilities.

Before this, the framework approach will be motivated.

2.1 State at the project beginning and requirements

At the beginning of our project in 1997, there wasn't a suitable solution available. The systems [21] [4] employed at this time were not able to provide high quality images of digitized handwriting. Major drawbacks were caused by only partially eliminated noise signals, cases where a correct separation of handwriting from textured background were impossible (figure 1) and cases where parts of the handwriting were lost (figure 2), in particular those with low contrast to the background. These problems revealed the need for specific methods dealing with handwritten documents and for taking into account the typicality of handwritings. Such methods should be aimed at focusing on a true representation of the original handwriting, as it is required for forensic examinations. (Technical factors, however, limit the scope of true handwriting representation. Handwritten documents are scanned with 200-600 dpi, depending on the examination goals as well as

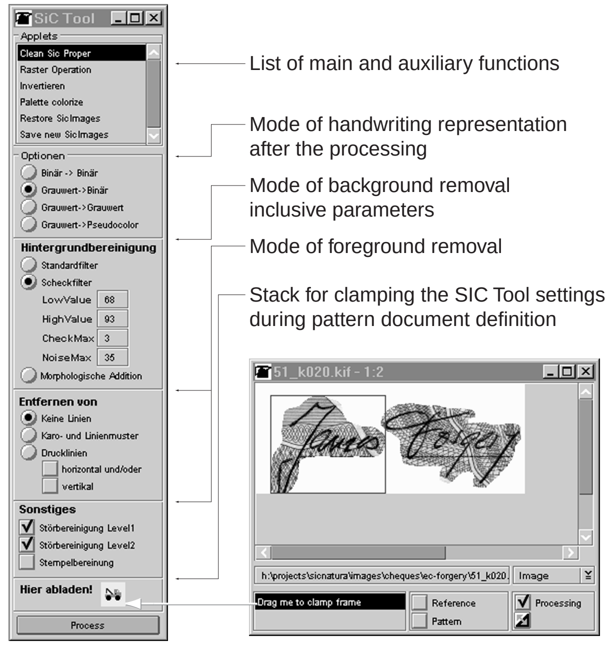

Fig. 3. A tool for specifying processing operators and their parameters

performance goals of the computer-based processing). Probably, a main reason for the mentioned problems is that, at the time, when the employed systems [21] [4] were developed, the technical premises were not yet available. (In the case of the FISH-system [21] the development started in 1975). Today, however, this is no longer the case. The time has come to consider the further development of traditional systems - reflecting on new technical possibilities.

Hence, the Bundeskriminalamt (the German central police bureau) initiated a research and development project. 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 [21] 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 [21] was conceptualized to handle 10.000 investigation cases per year. This means in the worst case there would be 10.000 different types of document. There are no restrictions to the document types. The only thing what they have in common is handwriting on them. 77.000 documents were stored until 1995 in the FISCH-system [21] 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.

2.2 System architecture

The description of the user interface is limited to such elements that enables the adaptation of the operators and parameters and that support the archiving of control sequences in

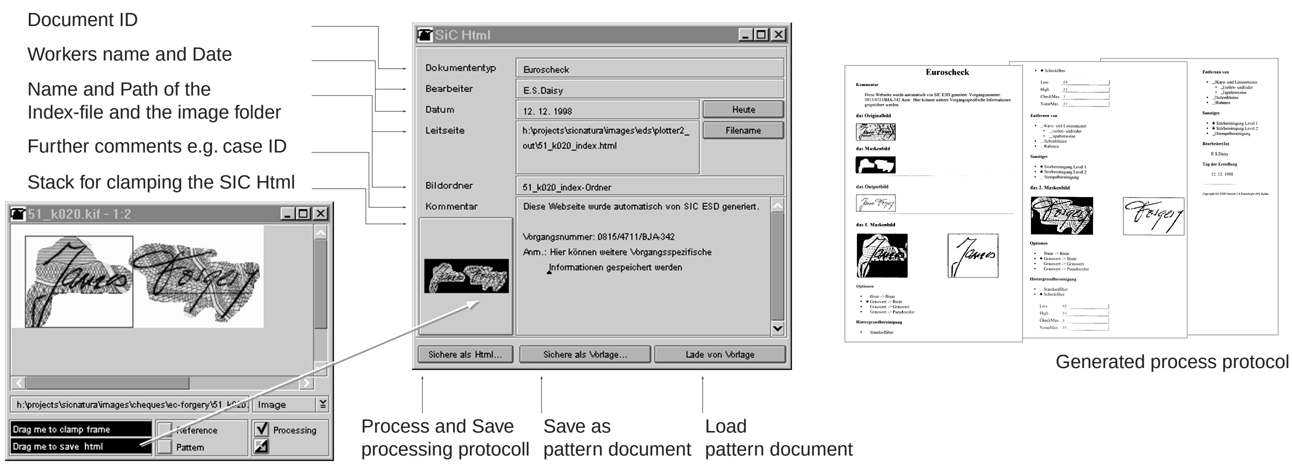

Fig. 4. Tool for using Html to store process protocols

such called *pattern documents* as well as of processing protocols to match the required proof of processing and validation.

The *SIC Tool* is the central element of the user interface. It supports the operator and the numerical parameter specification for document processing. From the chosen operators the resulting processing algorithm is derived automatically. Within the off-line phase of the framework application (compare section 2.3.1) the *SIC Tool* is applied for the manual definition of pattern documents as well as for the interactive processing of questioned documents without a specific pattern document. In the case of the application of additional structural parameters, like a mask image that defines various regions of interest within the questioned document, the *SIC Tool*-settings were applied sequentially and by simple drag-and-drop clamped to each region. In such a case, the management of this pill of various settings is carried out by the designed document class; for the *SIC Tool* it is not relevant. As it can be seen in figure 3, the *SIC Tool* enables the selection of various operators for back- and foreground removal. Among them are operators for document-independent as well as document-dependent processing (compare section 3). The document-specific processing (also section 3) is supported by the before mentioned mask images as well as by the extension of the function-list at the top of the *SIC Tool* (see section 3.3).

Either the created pattern document or the processing protocol of interactive processing can be stored by employing the *SIC Html*-tool (figure 4). The *SIC Html*-tool uses **h**yper**t**ext **m**arkup **l**anguage (html) to relate ASCII-data that specify the operators and the numerical parameter to images, like the questioned document image or the mask image. Moreover, the Html-standard was chosen to display and to exchange protocol files and pattern documents very easily by a usual web-browser. A proprietary editor is really not state of the art. Also, the html-files might be provided via Intranet within a lab, e.g. for validation purposes or for expert ratings. The *SIC Html*-tool supports also the inclusion of administration data, like the case number, the name of the expert and the date of the file creation. This information is important to meet the requirements of quality management in forensic handwriting examination.

For fundamental digital image processing, like the definition of the mask images, common tools should be used, too.

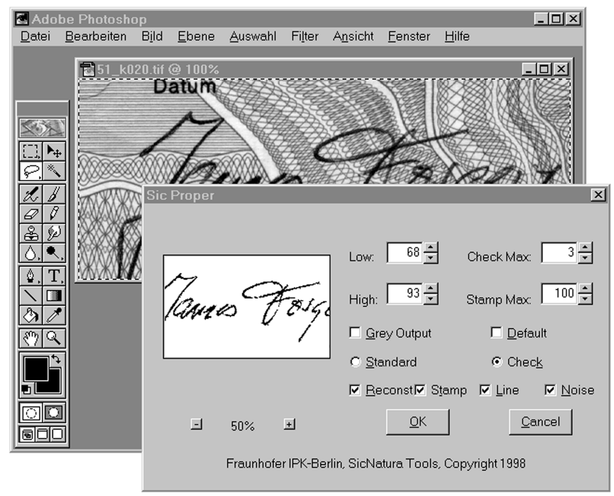

Fig. 5. Specific methods, taking into account the typicality of handwritings, realized as *Plug-In* for Adobe^{TM} Photoshop^{TM}

Within the forensic labs Adobe^{TM} Photoshop^{TM} is widely in use. So, some routines of inter-application communication support the data exchange between Photoshop^{TM} and the presented framework. Moreover, a *Plug-In* was developed to extent Photoshop^{TM} by sophisticated methods in handwriting pre-processing (figure 5). Hence, the forensic experts might tickle around with the provided document-independent operators, e.g. for parameter adaptations or the *Plug-In* might be employed in small-sized forensic labs that are not able to install the whole system.

2.3 Framework for document processing

From the authors point of view the realization of fully document-independent handwriting processing will stay a scientific challenge for a while. Although, there is a universal processing flow, which comprises document-layout recognition and document segmentation, handwriting segmentation and handwriting quality enhancement; a standardized queue of processing stages can not be used as a framework concept. With respect to the expected amount of different document

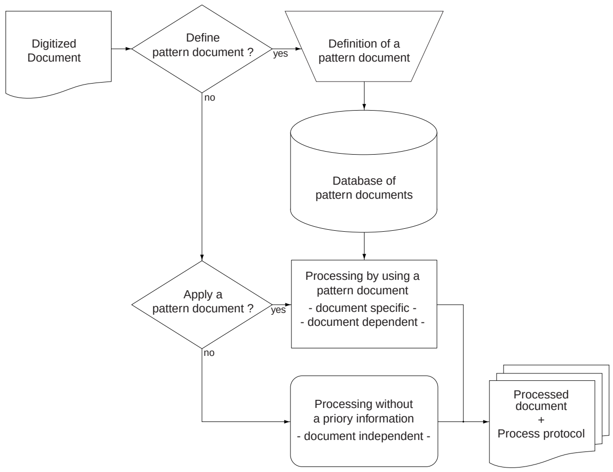

Fig. 6. Flow chart of document processing

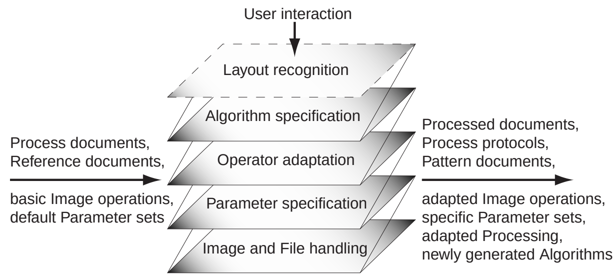

Fig. 7. Layer model of the framework for document processing

types, being subject of forensic analysis, and all those inhomogeneous structures and characteristics of the documents, the classical serial processing has to be replaced by an iterative and parallel one. Hence, the processing operations have to be selected by considering the prevailing document and by considering the specific processing-target. Moreover, an iterative and parallel processing concept allows taking into account intermediate data to improve final processing operations. It is understood that fixed parameter sets for the processing are not useful, too. And, even the number of different image operations should be extendable, because there are many approaches promising good processing results [6] [17] [18] [3] [28] [27].

The best solution seems to be an open layered framework concept to realize an assistance system in forensic handwriting analysis. It has to comprise a functional kernel providing basic processing operations for the most frequent document types as well as opportunities to adapt/extend the functionality by user interactions. The open layered architecture of the framework that was developted by the authors can be seen in figure 7. Each layer stands for an abstraction level of procedures working on the documents. Also, the layers support the understanding of how the framework operates and how user interaction might adapt/extend the functionality to specific demands.

2.3.1 The application of the framework

The framework can be applied in two separated cycles (see figure 6); once a pattern documents is prepared, it can be used for automatic controlling later on.

The such-called on-line phase 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 questioned document(s) correctly. The information stored in the pattern document include 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 might be processed interactively. In every case a protocol will be provided 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 were not predictable during the framework development. A detailed description of all the layers supporting user interaction in the off-line phase follows.

2.3.2 The layered model architecture

The presented framework, whose block diagram is shown in figure 7, 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 the huge amount of different documents the *LR* (layout recognition) layer is not working autonomously. Here, the user's assistance is still required.

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 (see section 3.3.1), 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.

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 [15] [10] (see figure 18 and section 3.3) 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 can verify the filters generated by the LUCIFER-module. A detailed description is given in section 3.3.

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

Even if the framework will be extended, this procedure will hold. The open layered architecture of the proposed framework is able to consider upcoming operators, too. Within each layer of the framework new approaches might be included and the users of the framework might interact with them. Algorithms of document processing, which are included within the framework up to now, are presented in section 3.

3 Methods for document pre-processing

The operators for document processing can be grouped into algorithms for document-layout analysis, like automatic location of text blocks; algorithms for background texture and/or image cleaning, as histogram and threshold based techniques; algorithms for foreground cleaning, as guideline and pre-printed data removal; and algorithms for handwriting reconstruction and enhancement. There are different strategies for the cleaning of background and foreground. Some approaches uses separated processing [5] [18] [6] [2] others use combined cleaning operators for fore- and background [20] [17].

Although, this grouping is fundamental, the authors want to focus on a further aspect. As mentioned before, it is necessary to handle various distinguished document types, therefore distinguished amount of a-priory knowledge about the characteristics of a document type is expected. Hence, the operators for document processing must be able to handle this different a-priory knowledge. Given information not to consider and operators and parameters not to adapt may lead to a document quality that is just moderate. To expect too much information could mean that some documents might not be processed. The authors propose a grouping of the processing operators that is derived form the given a-priory knowledge of a document type.

In general the document types being examined in the daily forensic casework can be 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, e.g. a drivers license, a pass document or a standardized bank check. The second document group includes such documents, which occur more or less often, such as

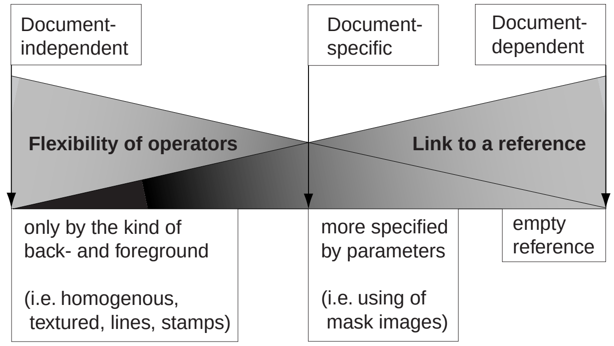

Fig. 8. Types of document processing

a registry-paper of a certain hotel or an insurance contract. 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 could be examined repeadly. The third document group covers all that document types appearing only few times and/or there is no way to get an empty reference, e. g. an aged testament or a piece of paper that was washed. From these three document groups the required operations were derived.

The processing operators and the derived generic types of document processing differ in their consideration of a-priori knowledge (e.g. empty reference, layout information and/or structural and numeric 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 a-priory knowledge)
- document-specific (by using adapted parameters)
- document-dependent (by using an empty reference)

The required processing quality of the handwritten documents and the flexibility of the whole framework are opposing targets (see figure 8). Therefore, the operators to be included into the framework have to be selected carefully. Also, there could be diverse operators available, which may be chosen alternatively. The processing 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 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 operators. Moreover, an embedded module (see figure 18) allows the automated generation of document-specific background filters. The document-dependent processing is realized by using the extented approach of morphological subtraction [20] [10]

3.1 Document dependent processing

Usually for the removal of nearly homogenous backgrounds, different kind of histograms and threshold techniques are used. Against it, it was pointed out by Okada and Shridhar [20] that enhanced inter-image subtraction, the so-called morphological subtraction, provides acceptable cleaning results for

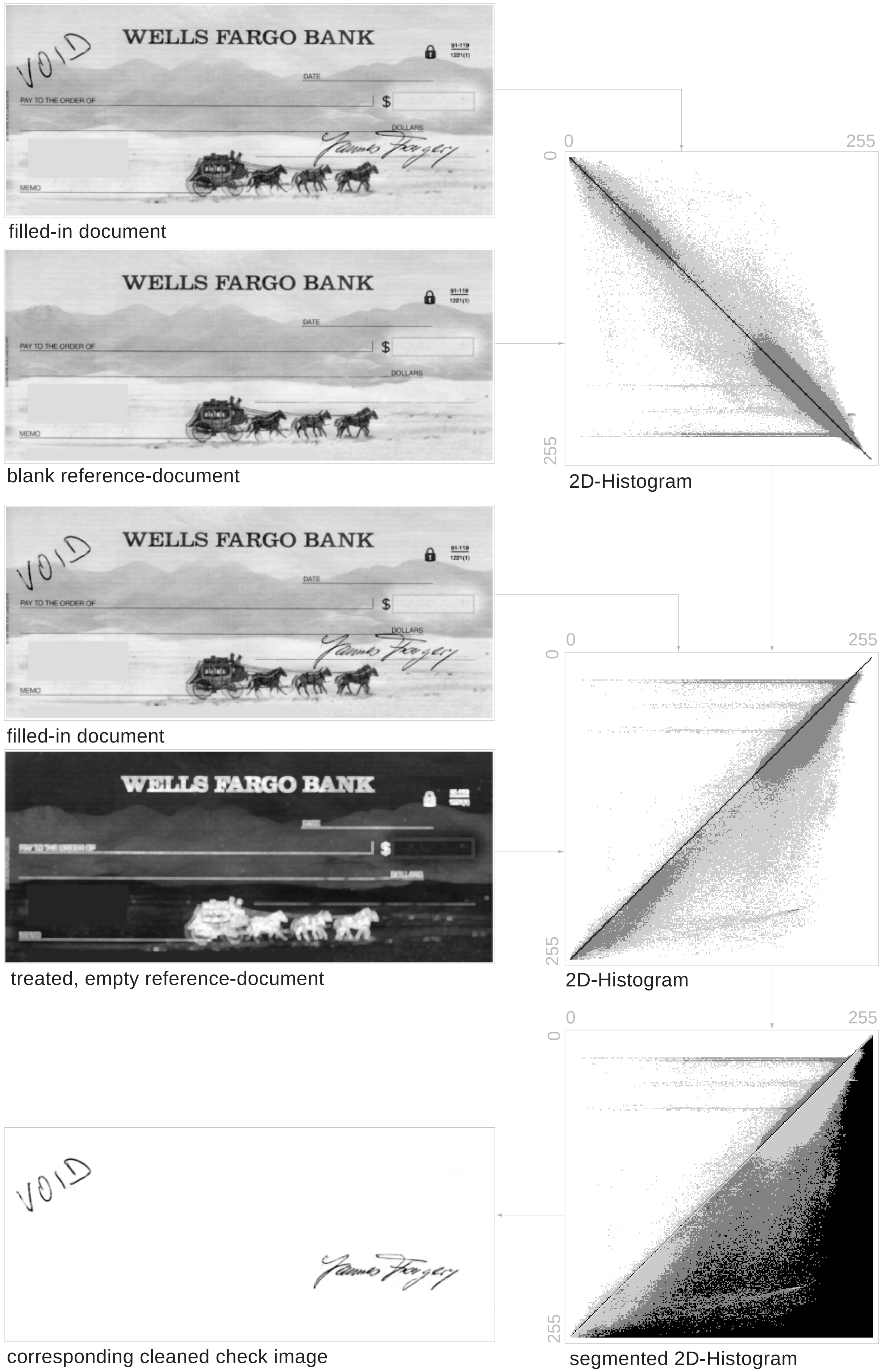

Fig. 9. Application of the 2D-Lookup for document-dependent handwriting segmentation

textured and/or image check backgrounds. For document-dependent processing, the authors provide a method for background and foreground cleaning [10] that was derived form the morphological subtraction. In contrary to the fundamental approach, a blank reference-document is filtered only once in the off-line phase and is used for grayvalue comparison during the background cleaning process [10]. For the comparison, a segmented 2D-Histogram is used for looking-up. Hence, it is possible to provide qualitatively comparable processing results by using a smaller amount of computational time. Concomitant to the introduction of 2D-Histogram segmentation, which will be referred to as 2D-Lookup (for further details see the appendix), the basic question is about the origins of the two images, which are used for the look-up. So far, they have been the treated, empty reference-document and the filled-in document, usually untreated (figure 9).

3.2 *Document independent processing*

3.2.1 *Textured background removal*

The processing of textured documents was studied intensively in the past [11] [10] [9]. Textured background removal on grayvalue images is a challenge. Handwriting might not be separated by global grayvalues intensities or empty reference subtractions [10]. Only, the consideration of local structures and local intensities by applying adapted morphological operators provided acceptable results (compare figure 10). The first implementation of this filter was done empirically by the authors, later it was proven by the LUCIFER-module [10] (section 3.3). Also, it was shown [10] that the 2D-Lookup can be derived as an abstraction of the former inter-image subtraction, which gives a more flexible approach to inter-image filtering. This is caused by the fact that for the graylevel lookup a blank check image is not mandatory. It is also possible to use two different filtered images of one filled-in check image.

In this manner, the difficult procedure for positional adjustment between filled-in and blank document image can be avoided. The fundamental problem for a practical and efficient usage of the 2D-Lookup is the selection of filter operations, which provide significant differences between background and user entered information. In an intermediate step, the 2D-Histogram has to be segmented. The result of this histogram analysis depends on both, the free-to-choose image operators and the quality of the labeling in the histogram. The application of the 2D-Lookup, as it is employed for textured background removal of various types of textures, like Eurocheck and American Express Traveler Check, is shown in figure 10. It has to be considered that in this application-case of the 2D-Lookup only such image elements can be eliminated, which are included in the labeled segment of the 2D-Histogram. Further image elements like the guidelines that show comparable characteristics to the handwritings (e.g. the same line thickness) and that is therefore not influenced by the filtering, has to be eliminated by an additional foreground cleaning.

For any further requests, like background removal of a new passport, applying the LUCIFER-module (compare section 3.3) to determine a suitable filter operation seems to be more convenient.

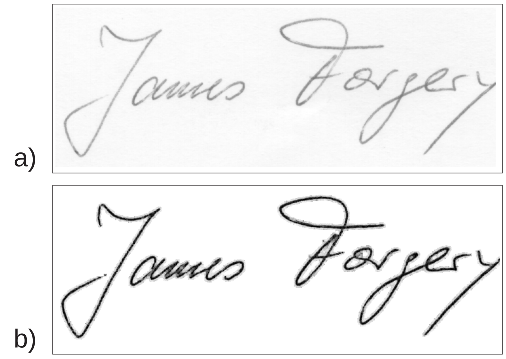

Fig. 11. Local contrast enhancement: (a) original signature image, (b) signature image after procesing

3.2.2 *Homogenous background removal*

In the following, methods for the elimination of homogenous background will be listed in detail. Independent of the writing material (e.g. kind of ink, kind of paper and paper color) and writing pad, these methods offer a binary image of high quality, without neglecting relevant parts of the writing product. It is expected that background has no textures and that there is a small contrast difference between handwriting and background. There are two major steps included in the procedure, which were optimized for the special case of handwritings.

Local contrast enhancement: Local image operators, along with other operators of mathematical morphology, are used for the purpose of contrast enhancement. In this context "local" means that the parameter assessment of the image operators is performed in the neighborhood of every pixel. This is done by a mask-like structuring element, which is positioned on every image position (slides over the image). The parameter choice of every position depends on a computation from the other image points underneath the mask. This procedure allows a better representation of local disturbances of the intensity distribution. Moreover, this method reduces the possibility of error resulting from global image properties such as a dark background causing the loss of low-contrast strokes.

The size of the structuring element should not exceed the average stroke width of the writing trace. Previous research [11] established that the stroke width of handwriting in a document for a broad range of varying writing devices such as ball point pen, fountain pen, or roller pen, digitized with a resolution of 300 dpi, was about seven to ten pixels. Due to this fact, for the contrast enhancement five pixels are chosen for the structuring element.

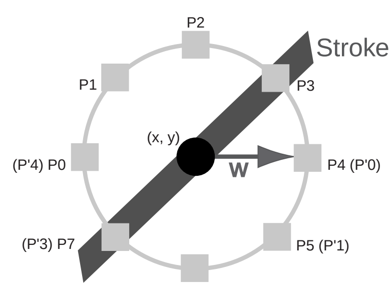

Fig. 12. Functional principle of the locally adaptive binarization [14]

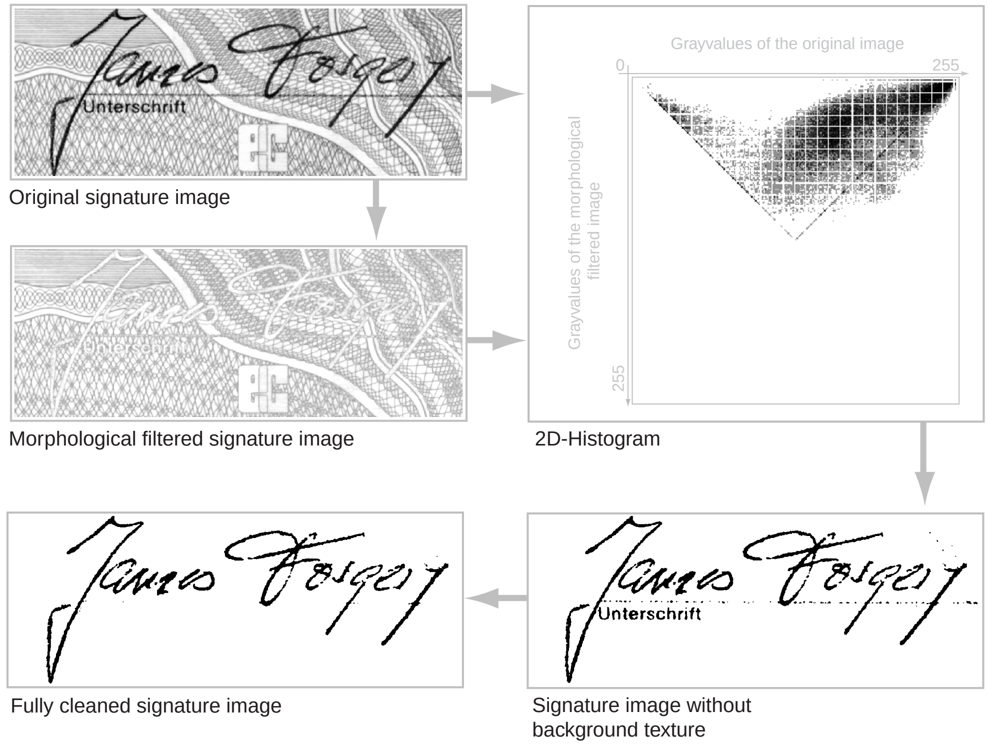

Fig. 10. Application of the 2D-Lookup for document-independent textured background cleaning

Figure 11 shows the result of the application of contrast enhancement and grayvalue normalization for handwriting. This time, even fine small details are preserved in the resulting image. Hence, by applying local-adaptive contrast enhancement, the inclusion of fine details is ensured. A detailed description of the operator was stated elsewhere [9].

Local adaptive binarization: The following binarization step should not result in broad lines or unexpected alterations to the fine lines in the original image. For this reason, a direction-oriented binarization, which considers image contents as well as grayvalue contrast, is advised. For the special case of handwritings, the slant and the width of every stroke are also considered. This is performed by local image operators with the size of the structural element of seven pixels.

$$F_b(x,y) = \begin{cases} 1 & \text{if} \bigvee_{\mathrm{i}=0}^{3}[\mathrm{L}(\mathrm{P_i}) \wedge \mathrm{L}(\mathrm{P'_i}) \wedge \mathrm{L}(\mathrm{P_{i+1}}) \wedge \mathrm{L}(\mathrm{P'_{i+1}})] \\ & \text{is true} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

with

w	pre-defined maximal stroke width
P'_i	$P'_{(i+4)\mathrm{mod}_8}$, for $i = 0, \ldots, 7$
$L(P)$	$= \mathrm{ave}(P) - g(x,y) > T$
T	pre-defined threshold
P_x, P_y	coordinates of point P
$g(x,y)$	grayvalue at point $P(P_x, P_y)$

This fundamental binarization process (compare figure 12 and equation 1) was introduced by Kamel and Zhao in 1993 and further extended by replacing an average filtering through a gaussian filtering [19].

3.2.3 Reconstruction and noise removal

A binarized image might contain distortions resulting from the poor quality of the document or the scanning process. The evaluation of a such called seed image, followed by a reconstruction of the binary image solve this problem. Here, a morphological reconstruction operation is employed [23]. The seed image is obtained by clearing all obscure image contents such as strokes with a width below a given threshold. Figure 13 shows the reference image of a handwritten signature. In the reconstruction step, all image segments, which do not contain at least one pixel of the reference image, are cleared. The result is a binary image of high quality (figure 13.) After these image operations, other algorithms are used to remove imprints like lines and machine printing. A further, more smoothly noise filtering is realized by the well-known analysis of connected components. The corresponding parameter controls the size of image elements to be removed.

3.2.4 Stamp and imprint removal

The removal of stamps and other machine-imprints employs an extended connected component analysis. Beside the segment-sizes, also, the frequency of occurrence in a row of uniform segments is considered. This approach works quite well (figure 14). However, problems arise if the stamp-segments are connected to the handwriting. Here, further research and development is required.

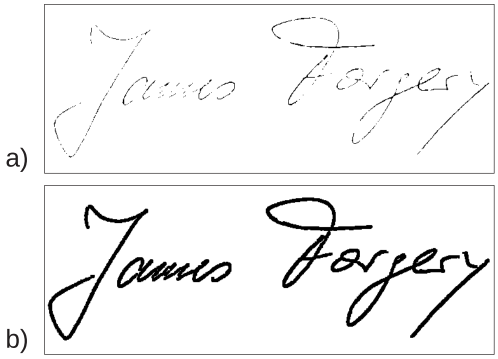

Fig. 13. Recontruction and noise removal: (a) seed image derived form the original image in figure 11 (b) final, high-quality binary image of the signature

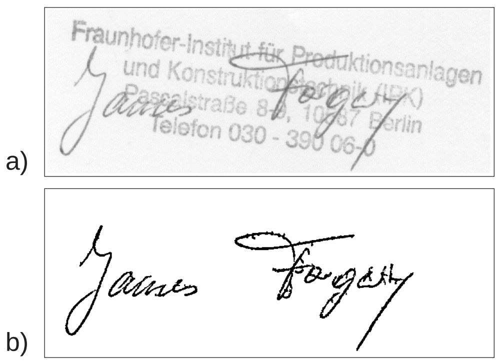

Fig. 14. Removal of stamps and other imprints: (a) original signature image with overlapping stamp (b) signature image after stamp removal

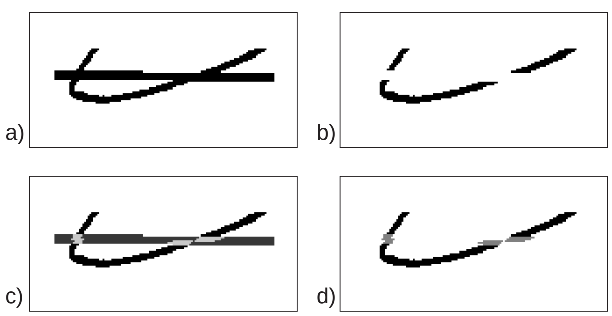

Fig. 15. Line reconstruction by using morphological filtering (length of the structuring element = 5 pixels; angle = $\pi/4$): (a) binarized image, (b) stroke image after line removal, (c) reconstructed stroke, (d) result image

3.2.5 *Line removal and stroke reconstruction*

There are two kinds of operators for horizontal and/or vertical line removal. The first one is applied on grayvalue images of handwritten documents. This approach is inspired by the legendary Otsu-method. Here, the operator acts like a background filter, by simple subtraction of the average pixelvalue per image row/column from the pixelvalue from the actual image position. To the rest of the work the filter for homogenous background removal is applied. This approach works quite well for lined and squared paper, but unsatisfying are the results for guidelines.

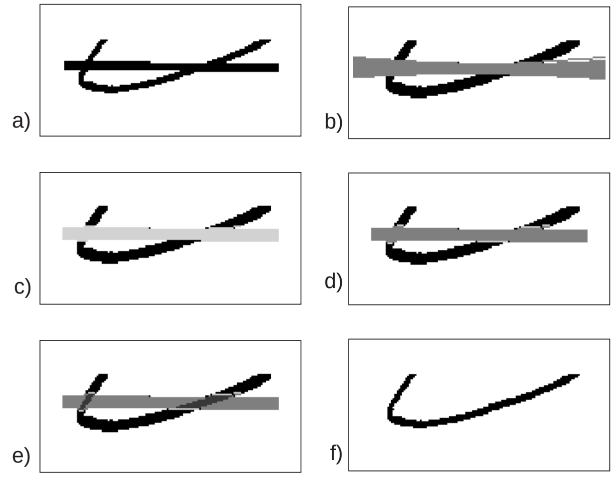

Fig. 16. Proposed line reconstruction: (a) binarized image, (b) searching for line candidates, (c) detected line, (d) line removal with concominant stroke marking, (e) reconstructed line, (f) result image

The second, more sophisticated algorithm is based on a structural analysis of the elements of the binary image. Most effort was spend in the reconstruction of the handwritten strokes during line-removal.

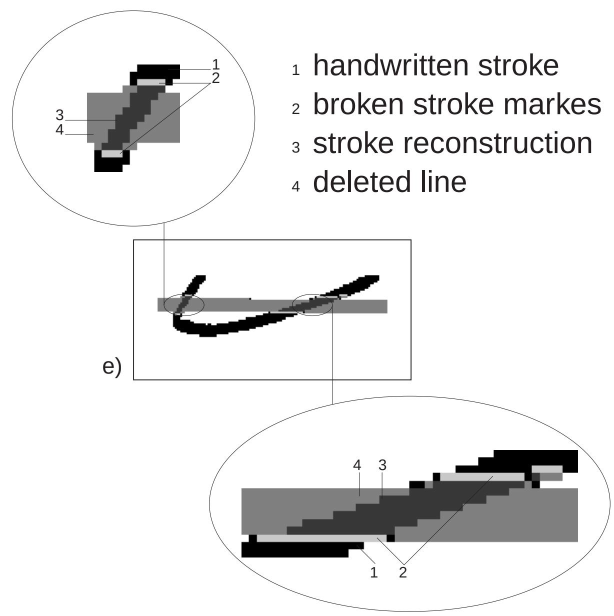

Fig. 17. Stroke marking: detail of figure 16

In contrary to the often stated assumption that the handwritten strokes may be reconstructed by applying morphological filtering, figure 15 shows that in the case of a small cutting angle and a thick guide lines the reconstruction fails. No matter whether binary or grayvalue morphology is used, if the cutting angle is smaller than the angle of structuring element (also called dynamic kernels [27]) the upper and lower stroke can not be connected. Furthermore, this procedure leads to distortion in the case of connected strokes.

The proposed method for stroke reconstruction combines handwritten stroke tracing and line removal in one step (fig-

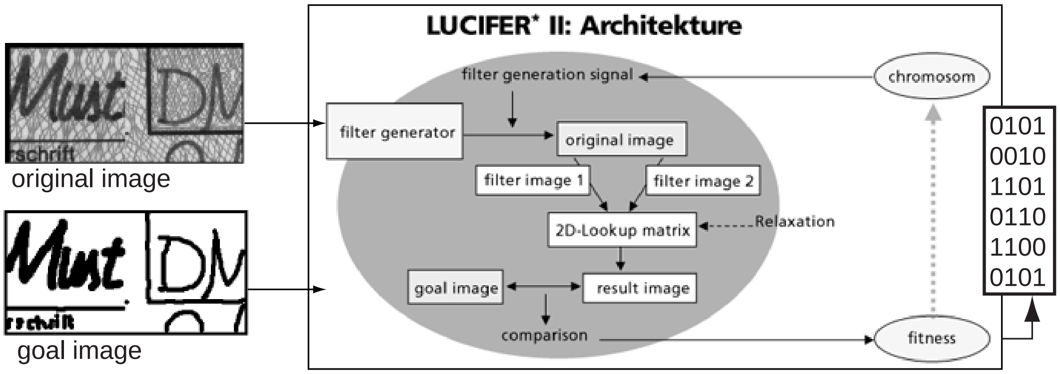

Fig. 18. Embedded LUCIFER-module for background filter generation

ure 16). After detecting a line within the image, it is tried to be removed carefully by checking the local neighborhood of the line for crossing strokes. All upper and lower crossing points were marked (figure 17). Than, the marks were grouped and straightly connected. The following look-up with the original image excludes stroke distortions. The results tune optimistically, however, in case of handwritten strokes that touch the line the grouping works just moderately.

3.3 Document specific processing

The document-specific pre-processing is realized into two manners. The first one uses the available operators. Then, depending on the users selection specific operators with their corresponding parameters were clamped to the pre-defined regions of interest. The second approach supports the generation of new, adapted background filters that takes the characteristics of a certain document type into account.

3.3.1 Application of mask images

To realize an efficient document processing the application of such called mask image is quite common. Such a mask image is employed to define regions of interest, like number-field or alpha-numeric-field, e.g. to support optical character recognition tools. Also, for the separation of handwriting this approach is useful. A document may comprise various regions that have quite different background textures like in the case of a passport. Then, a specific mask image helps to apply the correct filter operation to each region and to provide sophisticated cleaning results at the end.

3.3.2 Lucifer II framework

For solving the problem of determining a suitable filter operation that distinguishs between document background and user entered information, the LUCIFER II framework for filter design is used. LUCIFER II was developed for textural filter design in the context of surface inspection, and it uses evolutionary algorithms for its adaptation [15]. The feasibility of using the LUCIFER II framework for the design of specific background filters will be presented in the following.

General Overview: The framework is composed of (user-supplied) original image, filter generator, filter output images 1 & 2, result image, (user-supplied) goal image, 2D-Lookup matrix, comparing unit and filter design signal.

An evolutionary algorithm maintains a population of individuals, each of which specifies a setting of the framework. By applying the resulting 2D-Lookup and measuring the quality of coincidence of goal and result image, a fitness value can be assigned to each individual. These fitness measures are used for the standard genetic operations of an evolutionary algorithm. The 2D-Lookup algorithm, the fitness measure, the node and terminal functions of the individual's expression trees and the setting of the 2D-Lookup matrix will be shortly described in the next subsections. A more comprehensive introduction to the framework is given elsewhere [15].

Fitness function: A fitness measure is given by the degree of coincidence of goal image and result image. Both are binary images. The fitness measure, which is computed by the comparing unit, is a weighted sum of three single measures: the quota of white pixels in the result image, which are also white in the goal image (**whitegoalok**), the quota of black pixels in the result image, which are also black in the goal image (**blackgoalok**) and the quota of black pixels of the goal image, which are also black in the result image (**blackresok**). Note, that **blackgoalok** and **blackresok** are different. The multiple objective here is to increase these measures simultaneously. After performing some experiments with the framework, it was decided to use the following weighted sum for these three objectives:

$$f = 0.1\,\mathbf{blackresok} + 0.5\,\mathbf{whitegoalok} + 0.4\,\mathbf{blackgoalok}.$$

This fitness function was designed in order to direct the genetic search according to the schemata theorem, by which higher weighted objectives are fulfilled first.

Genetic Design of Filter Operations: One essential part of the framework are the two image processing operations. In order to generate them, genetic programming is used [16]. Genetic programming maintains a population of expression trees (shortly: trees). Every tree equals a program by its structure. For the design of the trees, the following operators were used as terminal functions: Move, Convolution, Ordered Weighted Averaging (OWA) [26], Fuzzy integral [12] [25], Texture Numbers. The node functions were out of the set of the operations minus, minimum, maximum, square, evaluate. In all tests, a maximum number of 50 generations was used. For

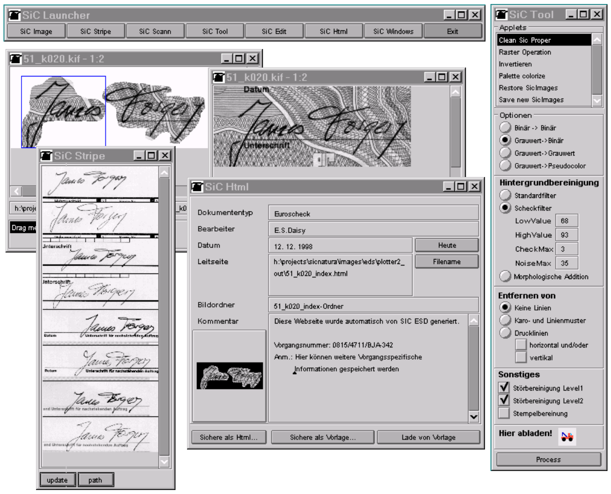

Fig. 19. Sneak preview of the major user interface

details, and also for the relaxation-based technique for the setting of the 2D-Lookup matrix, consider [15].

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, Windows 2000 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 (see section 2.3 and figure 19) was implemented in Smalltalk using ParcPlace Visual Works with C-Connect running on various platforms, too. Beside the kernel function calls and the inter-application communication, the management of process and mask images and the handling of the html-protocol-files are covered by the major interface. 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™ Photoshop™ *Plug-In* was implemented (compare section 2.2). With respect to the programmer interface of Photoshop™ the *Plug-In* covers only the document-independent processing functions for homogenous and textures backgrounds as well as for line, imprint and noise removal. Beside complete applications, the functional kernel was distributed as software library with programmer interface and now it is operating in automated check processing systems.

5 Discussion and results

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.

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.

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. Thats why, the computational effort for homogenous background removall is higher.

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.

6 Conclusion and further work

A framework was presented 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 and it is in use for more than two years now. 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 [15] [10] 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 [3] 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).

Also the authors wish to thank the unknown reviewers for their comments that helped to improve the papers quality.

Appendix: The 2D-Lookup Algorithm

The 2D-Lookup algorithm stems from mathematical morphology [22], [13]. It was primarily intended for the segmentation of color images. However, the algorithm can be generalized to be used for grayvalue images as well.

For starting off the 2D-Lookup algorithm, the two operation images 1 and 2, which are of equal size, need to be provided. The 2D-Lookup algorithm goes over all common positions of the two operation images. For each position, the two pixel values at this position in operation images 1 and 2 are used as indices for looking-up the 2D-Lookup matrix. The matrix element, which is found there, is used as pixel value for this position of the result image. If the matrix is bi-valued, the resultant image is a binary image.

Let I_1 and I_2 be two grayvalue images, defined by their image functions g_1 and g_2 over their common domain $P \subseteq N \times N$:

$$\begin{aligned} g_1 : \quad & P \to \{0, \ldots, g_{max}\} \\ g_2 : \quad & P \to \{0, \ldots, g_{max}\} \end{aligned} \tag{2}$$

The 2D-Lookup matrix is also given as an image function l, but its domain is not the set of all image positions but the set of tupels of possible grayvalue pairs $\{0, \ldots, g_{max}\} \times \{0, \ldots, g_{max}\}$,

$$l : \quad \{0, \ldots, g_{max}\} \times \{0, \ldots, g_{max}\} \to S \subseteq \{0, \ldots, g_{max}\}. \tag{3}$$

Then, the resultant image function is given by:

$$\begin{aligned} r \ : \ & P \to S \\ r(x,y) = \ & l(g_1(x,y), g_2(x,y)). \end{aligned} \tag{4}$$

In standard applications, every grayvalue is coded by eight bit, resulting in a maximum grayvalue of 255. Also, the domain of the image function is a rectangle. In this case, the 2D-Lookup is performed by the following (object-oriented) pseudo-code:

```
for x=0 to img width -1 do
begin
   for y=0 to img height-1 do
   begin
      g1 = g1(x,y)
      g2 = g2(x,y)
      out(x,y) = l(g1,g2)
   end y
end x
```

To give a simple example for the 2D-Lookup procedure, $g_{max} = 3$ is assumed in the following. Let

$$g_1 : \begin{array}{|c|c|c|} \hline 0 & 1 & 2 \\ \hline 0 & 3 & 3 \\ \hline \end{array} \quad \text{and} \quad g_2 : \begin{array}{|c|c|c|} \hline 2 & 3 & 1 \\ \hline 2 & 3 & 2 \\ \hline \end{array}$$

be the two input images and the 2D-Lookup matrix be given by

$$l : \begin{array}{|c||c|c|c|c|} \hline {}^{g_1}_{g_2} & 0 & 1 & 2 & 3 \\ \hline 0 & 0 & 0 & 1 & 1 \\ \hline 1 & 0 & 1 & 2 & 2 \\ \hline 2 & 1 & 2 & 3 & 3 \\ \hline 3 & 2 & 3 & 3 & 2 \\ \hline \end{array}$$

Then, the resultant image is

$$r : \begin{array}{|c|c|c|} \hline l(0,2) & l(1,3) & l(2,1) \\ \hline l(0,2) & l(3,3) & l(3,2) \\ \hline \end{array} \quad = \quad \begin{array}{|c|c|c|} \hline 1 & 3 & 2 \\ \hline 1 & 2 & 3 \\ \hline \end{array}$$

Since the goal image is supplied as a binary one and in order to keep user instruction as simple as possible, the 2D-Lookup matrix used in the Lucifer-framework contains only binary entries Black (0) and White (1).

References

1. R. A. Brooks, *A robust layered control system for a mobile robot*, IEEE Journal of Robotics and Automation, 2 (1986), pp. 14–23.
2. M. Cheriet, J. Said, and C. Suen, *A formal model for document processing of business forms*, Montreal, Canada, 1995.
3. C. Cracknell and A. Downtown, *A color classification to form dropout*, in Proceedings 6th International Workshop on Frontiers in Handwriting Recognition (IWFHR), Tajon, Korea, 1998.
4. W. C. de Jong, L. N. K. van der Kooij, and D. P. Schmidt, *Computer aided analysis of handwriting, the NIFO-TNO approach*, in 4th European handwriting conference for police and government handwriting experts, 1994.
5. G. Dimauro, S. Impedovo, G. Pirlo, and A. Salzo, *Automatic bankcheck processing: a new engineered system*, International journal of pattern recognition and artificial intelligence, 11 (1997), pp. 467–504.
6. S. Djeziri, F. Nouboud, and R. Plamondon, *Extraction of items from checks*, in Proceedings of the 4th International Conference on Document Analysis and Recognition (ICDAR), Ulm, Germany, 1997.
7. D. Doermann and A. Rosenfeld, *Recovery of temporal information from static images of handwriting*, International Journal of computer Vision, 15 (1995), pp. 143–164.
8. B. Found, *The Forensic Analysis of Behavioural Artefacts: Investigations of Theoretical and Analytical Approaches to Handwriting Identification*, PhD thesis, LaTrobe University Bundoora, 1997.

9. K. Franke and G. Grube, *The automatic extraction of pseudodynamic information from static images of handwriting based on marked grayvalue segmentation*, Journal of forensic document examination, 11 (1998), pp. 17–38.
10. K. Franke and M. 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.
11. K. Franke, M. Köppen, B. Nickolay, and S. Unger, *Machbarkeits- und Konzeptstudie Automaisches System zur Unterschriftenverifikation*, tech. rep., Fraunhofer IPK Berlin, 1996.
12. S. Grossert, M. Köppen, and B. Nickolay, *A new approach to fuzzy morphology based on fuzzy integral and its application in image processing*, in Proceedings ICPR'96, Vienna, Austria, 1996, pp. 625–630.
13. e. J. Serra, *Image Analysis and Mathematical Morphology. Volume 2: Theoretical Advances*, Academic Press, London, 1988.
14. M. Kamel and A. Zhao, *Extraction of binary character/graphics images from grayscale document images*, CVGIP: graphical models and image processing, 55 (1993), pp. 203–217.
15. M. Köppen, M. Teunis, and B. Nickolay, *A framework for the evolutionary generation of 2d-lookup based texture filters*, in 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), T. Yamakawa and G. Matsumoto, eds., 1998, pp. 965–970.
16. J. Koza, *Genetic programming — On the programming of computers by means of natural selection*, MIT Press, 1992.
17. L. Lee, M. Lizarraga, N. Gomes, and A. Koerich, *A prototype for brazilian bankcheck recognition*, International journal of pattern recognition and artificial intelligence, 11 (1997), pp. 549–570.
18. K. Liu, C. Suen, M. Cheriet, J. Said, C. Nadal, and Y. Tang, *Automatic extraction of baselines and data from check images*, International journal of pattern recognition and artificial intelligence, 11 (1997), pp. 675–697.
19. L. Lohmann, *Bildanalysesystem zur robusten Erkennung von Kennzeichen an Fahrzeugen*, PhD thesis, Technical University Berlin, 1999.
20. M. Okada and M. Shirdhar, *Extraction of user entered components from a personal bankcheck using morphological subtraction*, International journal of pattern recognition and artificial intelligence, 11 (1997), pp. 549–570.
21. M. Philipp, *Fakten zu* Fish, *Das Forensische Informations-System Handschriften des Bundeskriminalamtes - Eine Analyse nach über 5 Jahren Wirkbetrieb*, tech. rep., Kriminaltechnisches Institut 53, Bundeskriminalamt, Thaerstrasse 11, 65173 Wiesbaden, 1996.
22. J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, London, 1982.
23. P. Soille, *Morphological Image Analysis: Principles and Applications*, Springer-Verlag, Berlin, 1999.
24. Y. Solihin and C. Leedham, *Preparing handwriting images for forensic examination*, in Proceedings of the Eight Biennial Conference of the International Graphonomic Society, 1997, pp. 125–126.
25. M. Sugeno, *Theory of fuzzy integral and its applications*, PhD thesis, Tokyo Institute of Technology, 1974.
26. R. Yager, *On ordered weighted averaging aggregation operators in multi-criteria decision making*, IEEE Trans. SMC, 18 (1988), pp. 183–190.
27. X. Ye, M. Cheriet, C. Suen, and K. Liu, *Extraction of bankcheck items by mathematical morphology*, International journal on document analysis and recognition, 2 (1999), pp. 53–66.
28. 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.