

PDF hosted at the Radboud Repository of the Radboud University Nijmegen

The following full text is a publisher's version.

For additional information about this publication click this link.

<http://hdl.handle.net/2066/64411>

Please be advised that this information was generated on 2017-08-10 and may be subject to change.

WANDA: A common ground for forensic handwriting examination and writer identification

Katrin Franke ^{1*}, Lambert Schomaker ², Christian Veenhuis ¹
Louis Vuurpijl ³, Merijn van Erp ³ and Isabelle Guyon ⁴

¹ Fraunhofer Institute for Production Systems and Design Technologies,
Division of Security and Testing Technology, Berlin, Germany

² Rijksuniversiteit Groningen,
Artificial Intelligence Institute, The Netherlands

³ University of Nijmegen,
Nijmegen Institute for Cognition and Information, The Netherlands

⁴ ClopiNet , Berkeley, USA

This paper presents the WANDA Workbench, which is an open framework for electronic data processing. The framework provides generic interfaces for 'plug-in' applications for graphical user interfaces (client desktop with client plug-ins) and processing modules (server with server plug-ins). The applied plug-in concept allows for the functional extension of the workbench without changing the framework. Moreover, for data modeling, messaging and system configuration the eXtensible Markup Language (XML) is implemented, which supports the interoperability with existing applications and allows for continuous adaptations to the evolving state of technology.

The current application domain of the framework is writer identification and handwriting examination as frequently used in crime investigation and prosecution. Dedicated plug-ins using digital image processing and pattern recognition establish objective measurements and feature extraction to promote reproducible analysis results. Moreover, based on domain knowledge, which was collected from forensic document examiners, the data standard WandaXML was designed that allows for the annotation, processing, journaling and storage of digitized questioned handwriting documents.

* Contact: franke@ipk.fhg.de

1 Introduction

The computer-based identification of a writer on the basis of a digitized piece of handwriting is a challenging task for pattern recognition. A number of systems have been in use in Europe, the United States, and Australia. However, most of these systems are becoming increasingly outdated. Latest research results in digital image processing and pattern recognition are not being considered in current practice. Moreover, common data standards, procedures, and generic system environments are lacking.

In order to establish common grounds for international exchange, a standardized approach is proposed for the storage and 'plug in' application of analysis and evaluation procedures in forensic writer identification [24]. The major objective is to create a usable and effective platform for forensic writer identification, in particular by means of:

1. *Standardization* of the data-format,
2. *Modularization* and extendibility of the system concept,
3. *Objectification* in measurements and reliability of analysis results.

A new writer identification system should at least be a viable solution for the next decade. It should allow continuous adaptations to follow the current state of technology and it should be extensible, keeping backward compatibility in mind. Moreover, the development of a system for forensic handwriting analysis has to be grounded in established forensic investigation methods. And, potential users of an upcoming system should be involved during the specification and evaluation phase.

In order to ensure a long-term operation, two stable standards should be the basis of the proposed approach: the programming language Java and the data interchange language XML (eXtensible Markup Language). Particularly the usage of XML [2] supports the interoperability of existing and future applications. The greatest advantage of XML as a data interchange language is that it is text-based and human-readable. Therefore, it is independent from a particular platform, a particular programming language, and a particular middle-ware. XML tags can be freely defined and adapted to special needs, which makes XML so powerful and so flexible that it can be considered for a wide range of purposes as, for example, system configurations, data transfer protocols, and rich documents.

In the following, we will sketch the application domain and the general usage scenario of a computer-based system for forensic handwriting

analysis and writer identification (section 2). Next, we detail the concepts of the framework design (section 3), and provide technical aspects on the system architecture (section 4). Finally, we outline the current application of the presented framework and provide pointers to more general approaches (section 5). Envisaged further developments will conclude the work presented here (section 6).

2 Application Domain

In classical forensic handwriting examination, a human expert compares handwritings on the basis of well-defined sets of properties [7,9]. Like other related fields of forensic science, handwriting examination is primarily based on the knowledge and experience of the forensic expert. Due to the problem of non-objective measurements and non-reproducible decisions, it was attempted to support traditional methods, like visual inspection and expert rating, by computerized semi-automatic and interactive systems [1,4,6,8,12,15]. The subject of computer-based forensic handwriting examination are human handwriting samples, which might be digitized by using an optical device such as a digital camera or scanner, handwriting samples that are digitized by employing an electronic writing tablet or an electronic pen, but also multi-modally digitized handwriting, and signatures.

<i>Stage</i>	<i>Description</i>	<i>Processing mode</i>
1.a	case entry	interactive
1.b	scanning	interactive
1.c	preprocessing	semi-automatic
2.a	annotation	interactive
2.b	measurement	semi-automatic
2.c	feature extraction	automatic
3.a	normalization	automatic
3.b	comparison	semi-automatic
4.a	normalization	automatic
4.b	pre-selection	semi-automatic
4.c	identification matching	automatic
4.d	hit-list inspection	interactive
5	report generation	semi-automatic

Table 1: Stages in the forensic processing of handwritten samples. It should be remarked stages 3 and 4 can be alternative performed. While stage 3 covers procedures for verifying a questioned handwriting sample against available reference material, stage 4 is representing the identification procedures for resembling material out of a pool.

In the actual use of forensic handwriting systems, there is a clear work flow consisting of free interactive computer use as well as formalized, constrained working stages. Table 1 shows the stages in the work flow of handling forensic handwritten samples.

3 Framework Design

The concept behind the WANDA framework is inspired by two of insights. First, the need for longitudinal use of forensic annotated handwritten samples requires a stable and open data representation. The eXtended Markup Language XML has emerged as an ideal format. It is based on the elementary but convenient and legible XML text file. Formal syntax requirements allow for parsing and utilization of the XML content, depending on the changing application demands. This is opposed to long-term use of commercial and legacy databases, where the actual data formats are binary, obscure, and subject to version changes initiated by the software industry. Economic instability has revealed that a high dependence on industrial technology entails risks. Hence, within the WANDA framework XML shall be used.

The second insight is that it is desirable to frequently reuse/share used mechanisms in each new application. A set of software routines that cover fundamental mechanisms like accepting input from a user and responding by performing an action is also called an application framework [11]. For solving a specific problem, a framework needs to be extended according to one or multiple domain models describing the structure of the data and how it is processed. There are a number of open source and commercial software products that provide such called *Plug-In Interfaces*. To give an example, “ImageJ” and “Adobe Photoshop” are software programs for digital image processing, which functionality can be extended by so-called filter plug-ins.

These plug-in concepts were the source of inspiration for the WANDA framework design. Beyond that, WANDA should allow for generic data processing comprising data filtering, feature extraction, and classification. Moreover, the application domain, processing flow, and data structures should not have to be predefined. The framework with its domain-specific plug-ins might be tailored to tasks at hand, not to overwhelm the user and the user’s budget. At the same time, it should allow for several plug-in extensions in kind and in count.

4 System Architecture

A systematic overview of the WANDA system architecture is given in figure 1. The guiding idea is a rather strict separation of data, graphical user interfaces and processing modules. In order to support long-term interoperability between the diverse system components, data and process instructions are described in XML documents. System configurations and data transfer protocols are also XML-based [16]. The major components are: (a) clients with client plug-ins, which implement the graphical user interface, (b) servers with server plug-ins, which implement the processing modules, (c) data repositories and working sets, and (d) web clients for remote data retrieval.

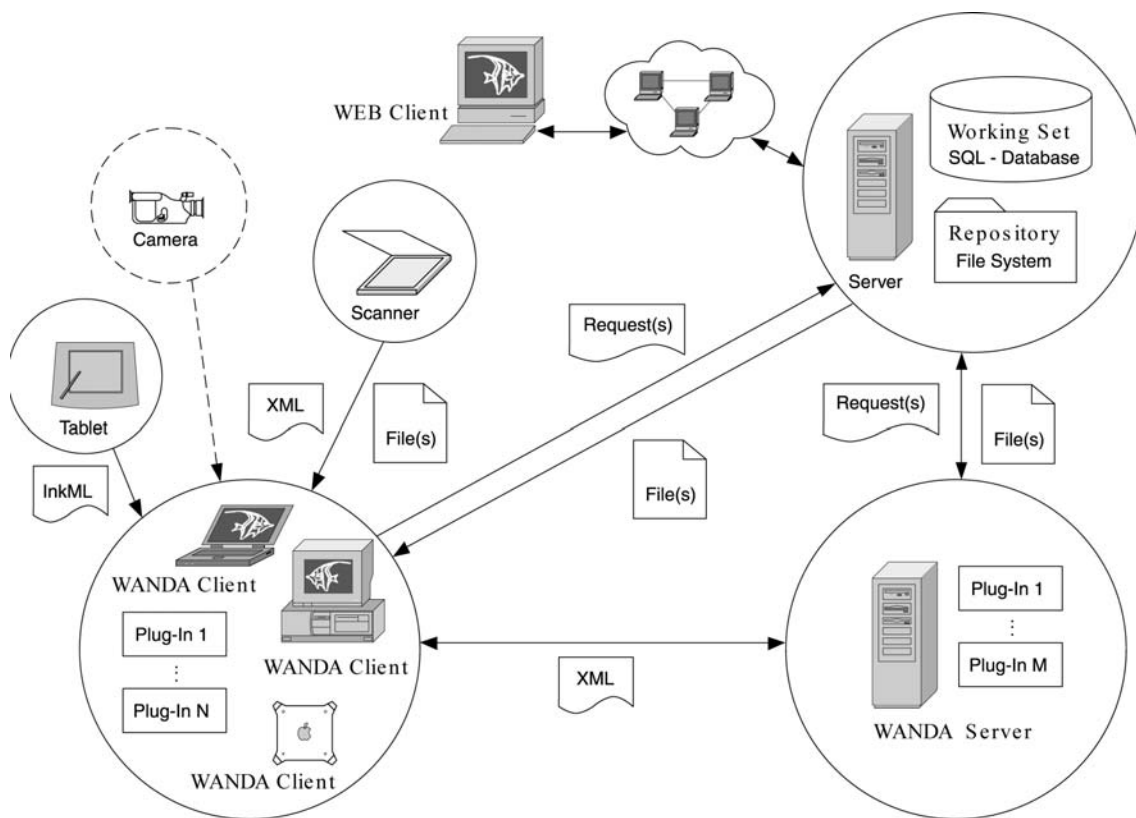


Figure 1: WANDA Workbench comprising: client(s), server(s), data repository, and web client(s).

4.1 WANDA Client and Plug-in concept for user interfaces

The WANDA client is an integrated desktop environment for all plug-ins that come with a graphical user interface [17]. This integrated desktop environment provides the user with a toolbar representing the available tools and a multiple document interface for displaying and interacting with these tools. Moreover, the client desktop acts as a manager for tools and provides them with: (a) Intercommunication (message exchange between the tools), (b) Access to commonly used functions such as network, printing etc. via centralized functions, (c) Window management (order/tiling/arrangement), and (d) Extended drag-on-drop functionality.

4.2 WANDA Server and Plug-in concept for processing modules

The WANDA server provides the functionality of processing modules (plug-ins) over a network [19]. By connecting to the server, clients can call upon these plug-ins via TCP/IP network sockets. Beside its functionality, the server plug-in's meta-data as for example name, description or version numbers, can be obtained. The server is multi-threaded. Thus, it can process any number of client requests concurrently. Each client gets its own corresponding thread on the server side and access to all installed plug-ins. The server organization is also inspired by the communication architectures between pattern recognition agents mentioned earlier [20]. A plug-in can be *any executable binary or script*, which can be written in any programming or script language. The plug-in might provide a single algorithm, complete methods or other auxiliary functions in the envisaged application domain.

4.3 Repository and Working Set

The Repository is a restricted file system where all data files are stored. The data files, as for example a digital image, are stored together with its XML-based description files, be it for annotation, features or processing journals. The file system area is intended to be exclusively accessed via a dedicated PHP Hypertext Preprocessor (PHP) script [13] that allows for administrating the restricted file system (create/delete/inspect directories) but also for up- and downloading files via the Hyper Text Transfer Protocol (HTTP).

During the development, it became apparent that the requirements for a fast and effective data entry selection cannot be met by the current XML software, i.e., at the scale which is required by the actual application domain (>>20.000 forensic cases may be present [12]).

The challenge then is to combine (a) the advantages of XML transparency and data handling convenience with (b) the advantages of traditional database methods [10,14], which excel in fast data handling and SQL-based logical subset selection. Consequently, procedures for transferring XML data into a traditional database are provided. Note: the usage of traditional databases is optional and not necessarily required for small and medium-sized applications.

In conclusion, the *Working-Set* concept allows for an exploitation of the best of two worlds. Rather than viewing the database as the oracle, which stores information for a prolonged period of time, the working set is a more volatile object, for fast and effective data-set selections. As database technology progresses, new database implementations may be adopted, with limited consequence for the overall software scheme. The open setup of the XML directory tree guarantees that the commitment of existing data to a new and possibly improved type of working set database is a feasible option.

5 Framework Applications

5.1 WANDA Workbench 2003

The realized prototype allows for fundamental processing of handwriting samples in the context of forensic handwriting examination and writer identification, particularly: Data acquisition (scanning, tablet input, combined); Preprocessing; Annotation of handwritten documents (in forensic context); Measurement of selected handwriting characteristics, Automatic feature extraction and Search in the data pool. In the following we describe those modules briefly.

5.2 Document Preprocessing

With respect to the expected amount of different document types being subject of forensic analysis, and all those inhomogeneous structures and characteristics of the documents, a strict serial processing has to be replaced by an iterative and parallel one [6]. Hence processing operations

have to be manually selected by considering the prevailing document and the specific processing target. These targets are for example interactive measurements, allograph shape examination or stroke-morphology analysis [18,30].

A basic set of processing operators is provided to the user. The selection of operators will allow for considering different amounts of a-priori knowledge (e.g. empty reference form, layout information and/or structural and numeric parameters). The following types of operators are defined, a more detailed discussion, however, is given elsewhere [6,29].

1. *Document-independent* (without priory knowledge) - Generic filters for the removal of homogeneous and textured background, lines, machine-print, noise and color dropout.
2. *Document-specific* (by using adapted parameters) - By considering additional structural and numeric parameters the generic filter operations might be adapted to special needs.
3. *Document-dependent* (by using an empty reference form). By considering a similar form without user-entered information image elements that belong to the form imprint is filtered by means of look-up techniques [6,29].

On the basis of these primary processing operations, a user can switch on/off single operations, like:

- Homogeneous background + gray value output image
- Textured background + lines + binary output image
- Color dropout + homogeneous background + machine print + noise + lines + binary output image
- Empty reference dropout (reference *ABC*) + noise + gray value output image

Even if the kernel functionality will be extended, this procedure will persist. Furthermore, specific designed processing operations might be stored and reloaded for repeated usage. The software operates on Regions Of Interest (ROIs). Particularly, the operators for background removal (e.g. homogeneous & textured background, color dropout) are capable of processing handwriting ROIs from page to allograph level. The operators for foreground removal (e.g. lines, machine print, noise) fully support the processing of handwriting ROIs from line to allograph. Few examples of handwriting and their preprocessing results are given in table 2.

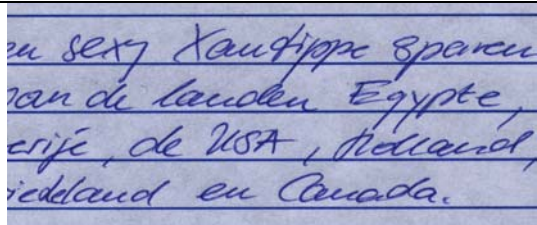
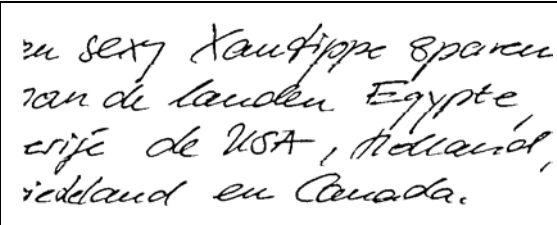




	
	
	

Table2: Results for document background removal and handwriting preprocessing.

5.3 Annotation

The proposed annotations aim at the establishment of a conceptional framework for dealing with the varieties of handwriting characteristics and their verbal descriptions as well as at the overcoming of lexical ambiguities and imprecision in forensic science.

In order to consider an investigation report as scientifically proved, the official expert report has to fulfill certain requirements. Therefore, experts from the field of forensic handwriting examination put a lot of effort in the harmonization and objectification of verbal categories on final conclusions of a handwriting's authorship. These activities can be further promoted if not only the final given evidence, but also the terminology for describing the investigation object and the supporting arguments during the validation on an hypotheses become standardized.

Moreover, the explicit specification of the application domain, is essential for the development and the use of intelligent systems. Not only to provide the system developer with the vocabulary for representing the domain knowledge; but, more importantly, to establish the envisaged inter-operation of humans (handwriting experts) and machines in daily forensic casework. For example, the current FISH system considers not only numerical handwriting features, but also very conventional nominal characteristics such as *writing style* or *connectivity* [7,12].

WANDA Annotation elaborates on the FISH implementation and introduces further categories and admissible values. Furthermore, a number of categories comprise values that enable the user to express a

degree of membership to a certain class/category. In this way, the introduction of sophisticated methods for computational intelligence is further promoted.

It has to be noted that the current realization of WANDA Annotation [5] is based on the knowledge of forensic handwriting experts working in Germany, combined with the expertise of researches from the field of handwriting analysis in Germany, the Netherlands and the USA. For the establishment of a widely accepted domain specification, the consideration of forensic handwriting expertise from different laboratories and countries is required. The presented version may motivate further discussions in this field.

5.3.1 Content Annotation

The content annotation is intended to describe content-related facts of a writing sample. The following list provides selected elements with their attributes and values.

<i>Document-Type:</i>	writing_product, note, letter, check, greeting_card, envelope, ...
<i>Document-Intent:</i>	message, bomb_threat, terrorism, racism, personal, public, ...
<i>Textblock-Type:</i>	whole_form, addressee_address_block, sender_address_block, zipcode, legal_amount, currency_amount, signature, ...
<i>Textblock-Length:</i>	character, word, few_words, line, ...
<i>Properties-Spelling:</i>	bad, ok, good
<i>Properties-Tone:</i>	kind, neutral, threatening, angry, ...
<i>Verbatim:</i>	

5.3.2 Material Annotation

The material annotation is intended to describe material-related facts of a writing sample. The following list provides selected elements with their attributes and values.

<i>Pen-Type:</i>	pencil, mechanical_pencil, ball_point_pen, roller_ball_pen, gel_ink_pen, steel_nib_pen, calligraphic_pen, ...
<i>Pen-Tip-Flexibility:</i>	low, normal, high
<i>Ink-Viscosity:</i>	low, medium, high
<i>Ink-Transparency :</i>	opaque, semi_opaque, transparent
<i>Paper-Type:</i>	writing, newspaper, concept, check, ...
<i>Paper-Absorbency:</i>	low, normal, high
<i>Pad-Hardness:</i>	soft, normal, hard
<i>Pad-Surface:</i>	even, structured

5.3.3 Script Annotation

The script annotation is intended to describe script-related facts of a writing sample. The following list provides selected elements with their attributes and values.

<i>Script-Type:</i>	arabic, chinese hanzi, greek, latin, russian cyrillic, ...
<i>Script-Language:</i>	german, english, french, dutch, ...
<i>Style-Major:</i>	cursive, mixed, handprint, blockprint
<i>Style-Connection:</i>	arced, garland, angular, straight, ...
<i>Style-Stroke-Quality:</i>	smooth, disturbed, highly disturbed

5.3.4 Writer Annotation

The writer annotation is intended to describe writer-related facts of your writing sample. The following list provides selected elements with their attributes and values.

<i>Person-Firstname:</i>	
<i>Person-Lastname:</i>	
<i>Person-Gender:</i>	male, female
<i>Language</i>	german, english, french, dutch, ...
<i>Education-Country:</i>	
<i>Education-Level:</i>	low, medium, high
<i>Properties- Handedness</i>	left-handed, unknown, right-handed

5.3.5 Investigation report

According to current forensic practice and the required quality assurance, so-called feature protocols have to be provided for each handwriting investigation/comparison. Those protocols are being supplied for the questioned handwriting sample as well as for the reference handwritings. Performing these procedures manually is extremely time consuming. Thus, interactive and semi-automated computer procedures might partly support the complete documentation of frequently needed aspects of handwriting.

For the semi-automatic generation of an expert report one can think of an easy-to-use graphical user interface where the expert only selects the specific categories and values that describe the handwriting product. On users demand the annotated data might be transferred into the case management system. Also, a text document could be generated, e.g. by means of XSL Transformations (XSLT) [3,21], which can be further

completed by using a standard text processor. In this way it will be possible to speed up the documentation of forensic evidences as well as to harmonize those documentation's.

Moreover, the internal representation of handwritings features, using a standardized XML format, supports the exchange of examination result between different laboratories and/or governmental entities. The proposed format will ensure that data can be rapidly imported into another computer system that does not need to be provided by the same manufacturer. So, WANDA promotes interoperability, and, with the anchors for further extensions, long-term usability.

5.4 Measurement

The WANDA Measurement plug-in (WAM) is a client side plug-in that allows the user of WANDA to interactively measure handwriting features in a selected ROI [23]. In this plug-in the user can select the measurement to perform, choose a letter to measure and is guided through the measurement process by means of step-by-step instructions. A special part of the WAM is the allograph matching interface, in which pattern recognition techniques are used to help the user find allograph prototypes of the questioned handwriting. The WAM offers the user a choice of ten measurements, which will be discussed next. In the following sections the allograph measure is explained in more detail, the preliminary validation of the plug-in is reported, and, finally, the WAM interface is shown.

5.4.1 The WAM measurements

The WAM has ten different measurements. The basic measurements consist of the various character heights (ascenders, descenders, corpus height, and height of oval characters), the slant of characters, and the character width. If present in the handwriting, the WAM allows the users to measure (upper and lower) loops of characters. As the only non-character based feature, the WAM also provides the measurement of the average distance between the baselines in a piece of handwriting. The last measurement, the new allograph measurement, is discussed in detail in the next section. These ten measurements are similar to the measurements performed in FISH (except for the allograph measurement). While the WAM was designed with the conclusions of the 'Vergelijk' project [15], the comparison between FISH [12] and the Dutch system SCRIPT [1], in

mind, the focus was strictly on FISH. There exist however a large overlap with the types of measurement from Script.

The ascenders and descenders are measured from the top to the bottom of the corresponding feature (not necessarily the top and bottom of the letter), the character height measures the vertical baseline to corpus line strokes in letters (e.g. in the 'n', 'u' and 'i'), while the oval height measures the distance of the top of the oval part of letters like 'a', 'o' and 'd' to the bottom of that oval. All measurements are from the first ink pixel to the last ink pixel (for example, see Figure 3a for a descender).

The line height, the distance between consecutive baselines, is measured by the user by taking the bottom of a letter on the baseline of every consecutive line that is included in the measurement (with a minimum of two). The average value of the distances between each consecutive pair of these points is displayed to the user as the measurement.

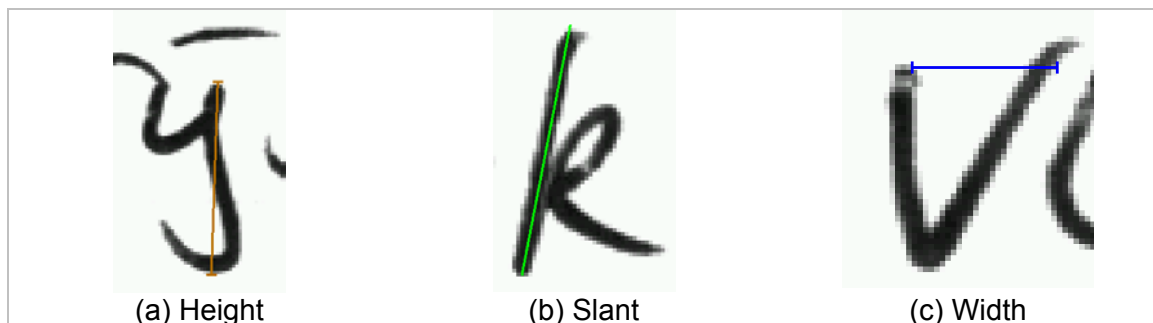


Figure 3: Typical examples of the height, width and slant measurements

The width of a character is measured on 'cupped' letters (e.g. 'u' and 'n'). It is the distance from the right edge of both cup-ends (i.e. the inside of the letter on the left side and the outside of the letter on the right, see Figure 3c). The slant (see Figure 3b) is performed by drawing a line on an ascender (preferably, though descenders are allowed). The angle between the drawn line and the x-axis is considered the slant of the letter. Finally, the loops are measured by calculating three values based on the trace of inside edge of the loop. By searching the inner bound of the loop, the WAM will determine the longest and shortest length across and return as measured values the longest length, the ratio between the longest and shortest length (the form of the loop) and the slant of the longest length line. These three values are used to characterize the loop.

5.4.2 Recognition-based measurements

In the WAM, a new measurement was introduced, the allograph match [28]. It finds, with help of the user, a prototype for a typical allograph that

was used in the handwriting, which is subsequently used as a feature in the identification process. With this measurement, the user is asked to trace a letter in the questioned document by pen-tablet or mouse. This creates an on-line trajectory of the allograph of that letter. The trajectory is then matched to a database of on-line allograph prototypes and the WAM will present the best matches for the user to choose from (see Figure 4). The label of this best matching prototype is then used as one of the features in writer identification.

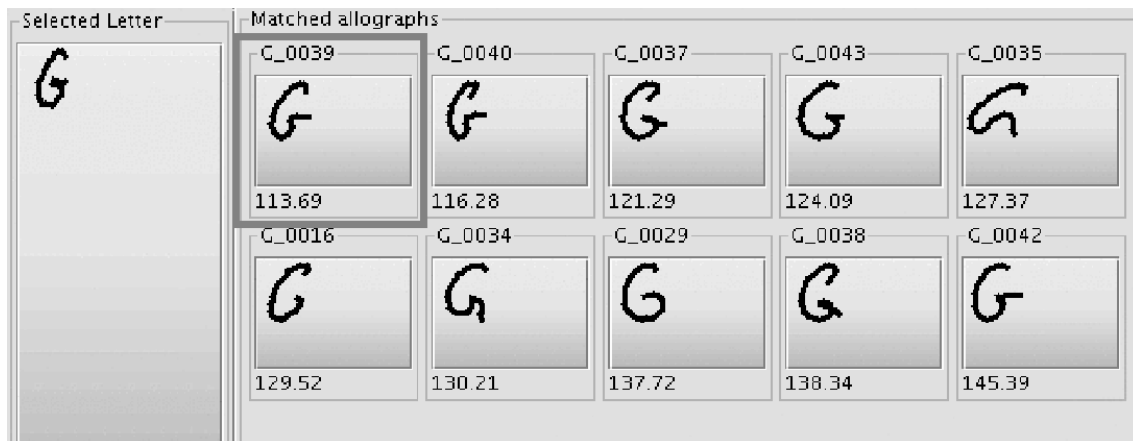


Figure 4: The presentation of the matching allograph prototypes.

The allograph match was added to the WAM as a proof of concept for recognition-based measurements. The idea is that if the program is able to recognize the written text, it can automatically and objectively measure all needed features for writer identification. As an added advantage, it opens up the well-researched area of pattern recognition techniques, techniques which are often directly applicable to writer identification. Another example of this can be found in [26, 27]. Note that the allograph matcher does not use any actual ink information, so it can only be used to provide a proof of concept for the second part of the recognition based measurements (from recognized text to measurements). However, it does provide the WAM with on-line information that the ink itself does not offer. Therefore, the allograph measure is, next to a proof of concept, a very useful measure by itself. As the allograph matcher is still a topic of research, it will regularly need an update. By making it a plug-in, the allograph matcher can be used already in WANDA without the need for issuing new versions to the users on a regular basis.

5.4.3 Validation of measurements

One of the hard requirements of the WANDA project was that the resulting measurements should be backward compatible with the FISH

measurements. FISH has been used for over a decade and a large database of cases has evolved. No compatibility would mean make this huge pool of knowledge useless. This was not acceptable.

As a result, the measurements from WANDA should closely resemble the measurements from FISH. A validation procedure was conducted to ensure that the results are compatible. Two kinds of validation were performed: one on the images that were measured and one on the measurement results. The first validation ensures that the scanned image on which WANDA measures the features is equal to the actual handwriting on paper.

The image validation was done by measuring features on paper and the same features from the image file. Measurements on paper were performed with both a measure with a millimeter scale (slightly too coarse) and a high-resolution microscope (very precise). Those were compared to the digitized version (300 dpi) of the image of which the number of pixels for the features was determined and to the results of measuring those features in the WAM (in tenths of millimeters). While 300 dpi is not the most precise resolution, it is equal or higher than the resolutions used in the FISH files (mostly 200 dpi). The resulting comparison showed that the image features were equal to the features on paper within the margin for error.

The validation on comparable results of FISH and WANDA was done on 11 scans provided by the *BundesKriminalAmt*. For each scan, the original FISH features were available. All FISH measurements were carefully redone in the WAM tool and the average measure was compared per measurement type (only averages for each kind of measurement for each image were available from FISH). These WAM too were comparable to the FISH-based data within the margin for error.

5.4.4 The WAM user-interface

The main window in the WAM (see Figure 5) provides an overview of all measurements that have been performed on the current Region Of Interest (ROI) that can be selected via the WANDA image browser. The measurements are color-coded by type, to enable the user to quickly identify them and to avoid cluttering the image too much. The buttons on the left allow the user to choose the type of measurement.

When the user initiates a measurement, a new window is opened. This window is similar for all measurements (see Figure 6). It consists of the ROI containing the handwriting to measure on (nr. 3 in Figure 6), a button

to select which letter is to be measured next (1), three control buttons (4, 5, and 6), and the instructions window (2). The novelty here is the instruction window, which gives clear instructions to the user on which step to take next in the measurement. This allows novice users to start measuring right away. A small usability study (5 novice users and two FISH/WANDA experts at the NICI), showed that the system was indeed easy to use for the novice users and offered fast shortcuts for the experts.

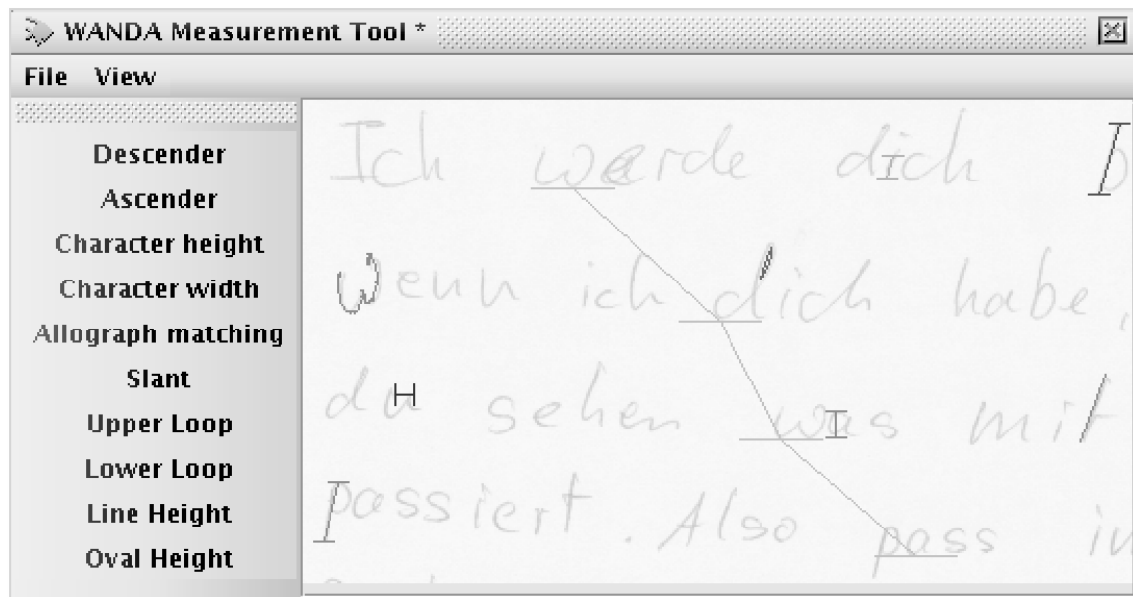


Figure 5: The main window of the WANDA measurement tool (WAM).

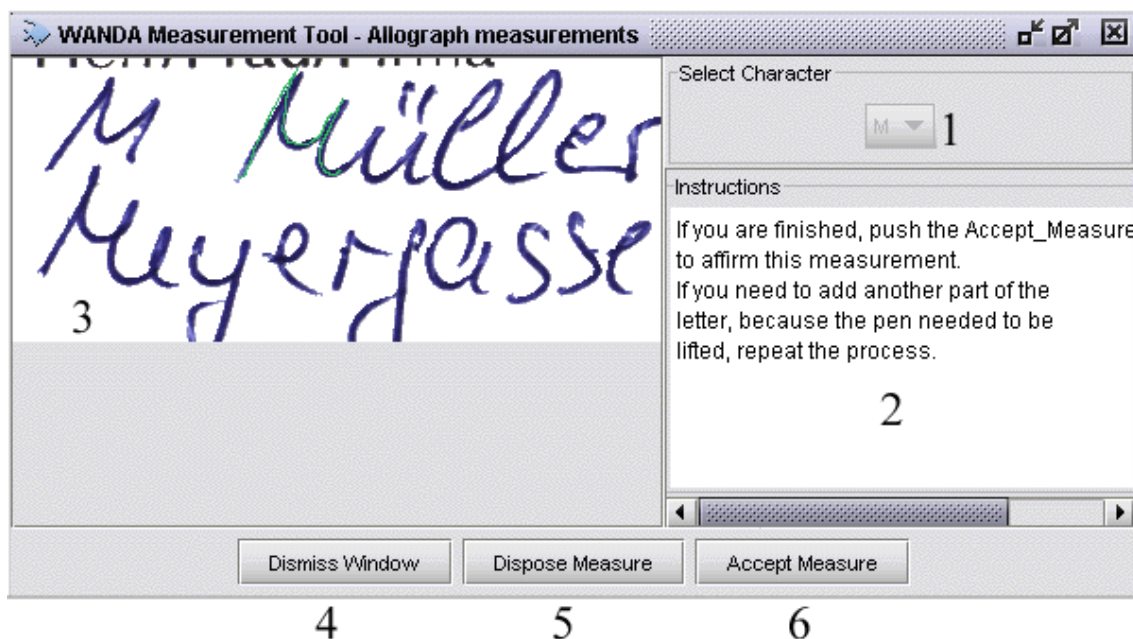


Figure 6: The measure window of the WANDA measurement tool (WAM)

The WANDA Measurement plug-in is an easy to use tool to interactively measure handwriting features. Especially the experiences of the novice users were important, as they clearly showed that the interface and measurement instruction window served their purpose.

Like WANDA, the WAM is a product that is continually being improved upon. A usability study will be held to find (and correct) those niggles that may still exist in the interface. Furthermore, the allograph matcher showed the interesting overlap between pattern recognition and identification. This cross-over can be exploited to further automatize the measurement process and to delete as much subjectivity in the results as possible.

5.5 Automatic feature extraction and writer identification

Traditional methods for forensic writer identification require considerable manual efforts in individual-character measurements by human experts. However, with current background removal methods, it now becomes feasible to use automatic image-based features on regions of interest which describe the individuality of handwriting style. Nevertheless, a single feature representation cannot be expected to capture all particularities of writing style, and combination methods are needed. The application domain precludes the use of training on the large data-sets such that sparse-parametric combination methods are preferred [25].

Forensic writer search is similar to Information Retrieval yielding a hit list, in this case of suspect documents, given a query in the form of a questioned script sample. Given the requirements, simple nearest-neighbor search is a viable solution. However, a proper distance function has to be identified. For the combination of results, rank combination (Borda) will be tested.

5.5.1 Feature groups

A number of feature groups (table 3) has been selected for this experiment, on the basis of literature and earlier work on on-line writer identification. Complementarity of extracted information in the feature group was an important design goal.

f1: ACF, auto-correlation function of the horizontal raster, detects the presence of regularity in writing: regular vertical strokes will overlap in the original row and its horizontally shifted copy for offsets equal to integer

multiples of the local wavelength. Every row of the image is shifted onto itself by a given offset and then the normalized dot product between the original row and the shifted copy is computed. The maximum offset ('delay') corresponds to 100 pixels. All auto-correlation functions are then accumulated for all rows and the sum is normalized to obtain a zero-lag correlation of 1.

	Feature	Explanation	Dim.	
<i>f1</i>	ACF	Auto-correlation in horizontal raster	100	Euclid.
<i>f2</i>	VrunB	PDF ¹ of vertical run lengths of ink	100	χ^2
<i>f3</i>	HrunW	PDF of horizontal run length of 'white'	100	χ^2
<i>f4</i>	Brush	Ink-density PDF at stroke endings	225	χ^2
<i>f5</i>	$p(\phi)$	Edge-direction PDF	16	Euclid.
<i>f6</i>	$p(\phi_1, \phi_2)$	Hinge angle combination PDF	464	χ^2
<i>f7</i>	$p(\phi_1, \phi_3)$	Horiz. edge-angle co-occurrence	512	χ^2
<i>f8</i>	WR	Writer: handedness, sex, age, style	16	Euclid.

Table 3: Feature groups used for writer identification and the used distance function $\Delta(\vec{u}, \vec{v})$ between two samples \vec{u} and \vec{v} .

f2: VrunB, PDF of vertical run lengths in ink

f3: HrunW, PDF of horizontal run lengths in background pixels. Run lengths are determined on the binarized image taking into consideration either the black pixels corresponding to the ink trace width distribution or the white pixels corresponding to the horizontal stroke and character-placement distribution for the writer. The histogram of run lengths is normalized and interpreted as a probability distribution. We use horizontal run lengths of up to 300 pixels (*f3*) and vertical run lengths (*f2*) of up to 100 pixels, i.e., the height of a written line in the data set used (resolution is 300 dpi). This feature is not size invariant. However, size normalization is not an issue in interactive writer search. The run-length PDFs provide orthogonal information to the directional features.

f4: Brush, ink-density PDF at stroke endings. It is known that axial pen force ('pressure') is a highly informative signal in on-line writer identification. In ink traces of ball-point pens, there exist lift-off and landing shapes in the form of blobs or tapering due to the ink-depositing process during take off and landing of the pen. A convolution window of 15x15 pixels was used, only accumulating the local image if the current region

¹ PDF – probability density function

obeys to the constraints for a stroke ending. This constraint is determined by a supraliminal ink intensity in the central pixel of the window, co-occurring with a long run of white pixels along minimally 50% of the perimeter of the window, which is interrupted by one ink strip of at least 5 % of the window perimeter. After summing all luminances, the accumulator window is normalized to a volume of 1, yielding a PDF for ink presence at stroke endings in any direction. This feature is not size invariant: the window of 15^2 pixels was chosen because it captures 6-7 pixel-wide ink traces (size normalization is assumed)

f5: $p(\phi)$ simple edge-direction PDF, is computed by considering the PDF of quantized directions of the Sobel edges in the image. Sixteen bins were used in the histogram (Figure 7).

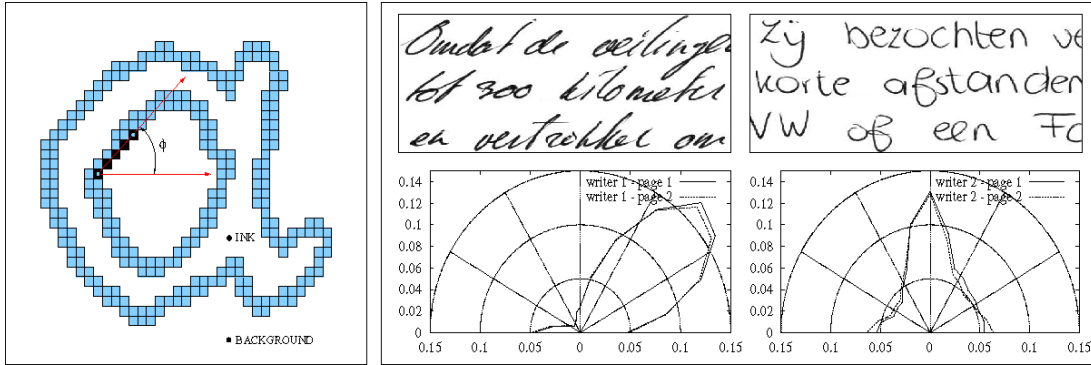


Figure 7: (left) Two handwriting samples from two different subjects. (right): We superimposed the polar diagrams of the edge-direction distribution $p(\phi)$ corresponding to pages 1 and 2 contributed to our data set by each of the two subjects.

f6: $p(\phi_1, \phi_2)$ hinge-angle combination PDF. In order to capture the curvature of the ink trace, which is very typical for different writers, another feature is needed, using local angles along the edges [26]. The computation of this feature is similar to the one previously described, but it has added complexity. The central idea is to consider the *two* edge fragments emerging from a central pixel and, subsequently, compute the joint probability distribution of the orientations of the two fragments of this “hinge”. The final normalized histogram gives the joint probability distribution $p(\phi_1, \phi_2)$ quantifying the chance of finding in the image two “hinged” edge fragments oriented at the angles ϕ_1 and ϕ_2 respectively. The orientation is quantized in 16 directions for a single angle. We will consider only the non-redundant angles ($\phi_1 > \phi_2$) and we will also eliminate the cases when the ending pixels have a common side. Therefore the final number of combinations is $C(2n, 2) - 2n = n(2n - 3)$ (464 dimensions).

f7: $p(\phi_1, \phi_3)$ horizontal edge-angle co-occurrence. This feature is a variant of the edge-hinge feature, in that the combination of angles is computed along the rows of the image. For the angle of a found edge fragment i , the co-occurrence probability is computed with the angles of fragments j which are horizontally displaced from i .

f8: Writer characteristics (WR) is a 'pseudo' feature vector, containing writer parameters which are often known in the application context: Style may be one of Handprint, Cursive or Mixed. The parameters are represented as a bit vector. This feature is added to underscore the possibility of using heterogeneous sources of information in a rank-combination scheme.

5.5.2 Feature combination schemes

Given a sample of unknown identity u and a universe of samples of known writer identity W , each uniquely labeled $[1 : m = |W|]$, and assuming there exist N feature groups describing a sample, we can construct a Borda rank-combination scheme. Assume a set of N distance vectors $\vec{\delta}_i = (u, W)$ between the unknown sample u and the reference set W for each feature group $i = [1, \dots, N]$, such that each dimension of the distance vectors corresponds to one and the same sample index. Furthermore, given that a vector of ranks will be denoted by \bullet , assume the availability of a rank operator $\dot{p}(\vec{x}), \vec{x} \in R^m$. which returns for each dimension in \vec{x} (i.e, handwritten sample) its unique rank in the set W with respect to u according to values in \vec{x} , in ascending order. The dimension $d(r)$ of a rank value uniquely refers to a sample in W . Thus $\dot{p}(\vec{\delta})$ guarantees that $(\Delta u, d(r_1) < \Delta u, d(r_2) < \dots < \Delta u, d(r_m))$, where $\Delta u, d(r_j)$ is the distance between an unknown sample u and a known sample of j^{th} rank, indexed $d(r_j) \in W$. Then a Borda rank combination scheme can be considered as a rank-combination function $\vec{\beta}()$ operating on a tensor:

$$\dot{L}_{Borda}(u, W) = \dot{p} \left(\vec{\beta} \begin{pmatrix} \dot{p}(\vec{\delta}_1(u, W)) \\ \dot{p}(\vec{\delta}_2(u, W)) \\ \dot{p}(\vec{\delta}_3(u, W)) \\ \vdots \\ \dot{p}(\vec{\delta}_N(u, W)) \end{pmatrix} \right)$$

where $\bar{\beta}()$ returns a vector in R^m which has a monotonous relation to the combined rank vector. The output hit list contains the samples in the final rank order \dot{L} . In the regular Borda vote, $\dot{L}_{Borda}(u, W) = \dot{p}(\sum_{i=1}^N \dot{p}(\bar{\delta}_i(u, W)))$, i.e., $\bar{\beta}()$ is the Sum function: ranks are summed per dimension before being resorted by $\dot{p}()$. However, many Borda-operator variants are known $\bar{\beta}()$: Sum, Max, Median, Min, Majority, Plurality etc. In this study, we tested the use of the Sum operator. The problem of the Sum operator is that all votes are treated equally. Recent tests with the Min operator, reported elsewhere [25, 27], have given indications that this rule may be preferable if a good solution for ties is used.

5.5.3 Data and Evaluation

We evaluated the effectiveness of different features for writer identification using the *Firemaker* data set [15]. A number of 251 Dutch subjects wrote four different A4 pages. On page 1 they were asked to copy a text presented as machine-printed characters. On page 2 they were asked to describe a given cartoon in their own words. The same kind of paper, pen and support were used for all subjects. The A4 sheets were scanned at 300 dpi, 8 bit / pixel gray-scale. Performance was tested using leave-one out. For a query sample, the set w will contain one matching sample of the same writer and 500 distractor samples by 250 other writers. Ongoing studies have revealed still better identification performances if (a) feature vectors are computed separately from upper and lower parts of lines of text [26], and additional improvement if (b) local component-shape features are used. A summary of results on writer identification experiments using the *Firemaker* data set is given in table 4.

System	$N_{writers}$	Style	Set	Amount	Top1	Top10	Ref.
SysA	100	mixed_lower	p1	page	34	90	[15]
SysB	100	mixed_lower	p1	page	65	90	[15]
Edge	250	mixed_lower	p1,4	page	33	73	[25]
Hinge	250	mixed_lower	p1,4	page	71	90	[25]
Brush	250	mixed_lower	p1,4	page	53	81	[25]
MiscBorda	250	mixed_lower	p1,4	page	79	95	[25]
splitEdge	250	upper_case	p2	paragraph	45	83	[27]
splitAla	250	upper_case	p2	paragraph	64	91	[27]
splitHinge	250	upper_case	p2	paragraph	78	95	[27]
splitEdge	250	upper_case	p1	paragraph	29	69	[27]
splitAla	250	upper_case	p1	paragraph	64	86	[27]
splitHinge	250	upper_case	p1	paragraph	79	96	[27]

Table 4: Results in % of correct writer in hit list of given size (1 or 10).

6. Future Work

There are many ideas for the further extension and application of the WANDA Workbench. Primary we are envisaging its usage in forensic case work. We expect a huge potential for research and development in the field of forensic handwriting and signature analysis as well as writer identification. Particularly, advanced software modules for the computer-based validation of signatures [4,31,32] and the analysis of stroke morphologies [18,30] introduces new objective examination procedures that are not available in current computer systems [1,8,12,15].

In the near future, we foresee adaptations and extensions of the data model and its corresponding modeling language WandaXML according to reviews by forensic handwriting experts as well as by new demands arising with upcoming investigation and analysis objectives [33,34,35]. It is understood that those objectives will also result in the integration of further client and server plug-ins. Moreover, the distributed client/server architecture allows for joining and sharing collected handwriting data, domain-specific knowledge and implemented software routines

Besides representing existing practices in forensic data analysis in a standard way, WANDA can facilitate experimenting with new ideas. The framework and data format will be particularly useful for research and development since it will allow researchers to parse and annotate large bodies of data, partially manually, partially automatically. Such data will lend themselves to statistical data analysis to automate writer identification, extract new features of interest, infer new correlation between handwriting attributes, and improve handwriting recognition. The existence of publicly available data formatted in a standard way will stimulate research for forensic applications. We foresee the possibility of organizing benchmarks based on such data. A standard data format also facilitates and stimulates data exchange. We anticipate that there will be a growing body of data incorporating contributions of many institutions.

Moreover, we are thinking of further improvements of the WANDA Workbench itself, as for example providing easy-to-use administration tools with graphical user interfaces and wizards. Another point is extending the workbench to customizable central views. It should be possible that developers can implement and integrate their own central views. This is the final step for allowing such called *domain theme* where the WANDA workbench can be set up to an application domain by using a dedicated data viewer e.g. video player with corresponding client and server plug-ins as for example for video processing.

Acknowledgments

Although many persons have directly or indirectly contributed to the framework design and implementation, special thank goes to those who were actively involved, particularly: Ottmar Bünnemeyer, Xiufen Liu, Mario Köppen, Tomas Kühn, Altug Metin, Martin Penk, Steffen Rose, and Christian Taubenheim (FhG-IPK), Marius Bulacu, Johan Everts, Maarten Jacobs, Geertje Zwarts (RUG), Hubert Voogd, Martijn Beenders (NICI), Stefan Giesler (SWE), Hartmut Gieschen, Manfred Hecker, Axel Kerkhoff, Werner Kuckuck, Manfred Philipp (BKA), Gerhard Grube, Reinhard Zschach (LKA Berlin).

This work was sponsored by the Bundeskriminalamt (the German Federal Police Bureau), section handwriting examination (B2.30-2212/02).

References

1. W.C. de Jong, L.N. Kroon-van der Kooij, and D.Ph. Schmidt. Computer aided analysis of handwriting, the NIFO-TNO approach. In Proc. 4th European Handwriting Conference for Police and Government Handwriting Experts, 1994.
2. eXtensible Markup Language (XML). <<http://www.w3.org/XML/>>, 2003.
3. eXtensible Stylesheet Language (XSL) - W3C Recommendation. <<http://www.w3.org/TR/xsl/>>, 2001.
4. B.J. Found. *The Forensic Analysis of Behavioural Artefacts: Investigations of Theoretical and Analytical Approaches to Handwriting Identification*. PhD thesis, LaTrobe University Bundoora, 1997.
5. K. Franke, I. Guyon, L.R.B. Schomaker and L.G. Vuurpijl. *WandaXML - A data standard for the annotation and storage of handwriting samples in the context of (computer-based) forensic handwriting analysis and writer identification*. International Unipen Foundation, <<http://unipen.nici.kun.nl/>>, 2003.
6. K. Franke and M. Köppen. A computer-based system to support forensic studies on handwritten documents. *International Journal on Document Analysis and Recognition*, 3(4):218--231, 2001.
7. M.R. Hecker. *Forensische Handschriftenuntersuchung*. Kriminalistik Verlag, 1993.in German.
8. S. N. Srihari, S-H. Cha, H. Arora and S. Lee (2002), Individuality of handwriting. *Journal of Forensic Science*, 47, 856-872.
9. L. Michel. *Gerichtliche Schriftvergleichung*. De Gruyter, 1982. in German.
10. MySQL. <<http://www.mysql.com/>>, 2003.
11. ParcPlace-Digitalk, Inc. *VisualWorks - User's Guide*, Rev. 2.0. 999 East Arques Avenue, Sunnyvale, CA 94086-4593, 1995.
12. M. 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, Germany, 1996. in German.
13. PHP Hypertext Protocol (PHP). <<http://www.php.net/>>, 2003.

14. PostgreSQL. <<http://www.postgresql.org>>, 2003.
15. L.R.B. Schomaker and L.G. Vuurpijl. Forensic writer identification: A benchmark data set and a comparison of two systems. Technical report, Nijmegen Institute for Cognition and Information (NICI), University of Nijmegen, The Netherlands, 2000.
16. Simple Object Access Protocol (SOAP) - W3C Note. <<http://www.w3.org/TR/SOAP>>, 2000.
17. C. Taubenheim and K. Franke. *Programmers Handbook WANDA Client*. Fraunhofer Institute for Production Systems and Design Technology (IPK), Division Security and Testing Technology, Pascalstr. 8-9, 10587 Berlin, Germany, 2003.
18. K. Franke, O. Bünнемeyer, T. Sy (2002). Writer identification using ink texture analysis, in Proc. 8th International Workshop on Frontiers in Handwriting Recognition (IWFHR), Niagara-on-the-Lake, Canada, pp. 268-273.
19. C. Veenhuis and K. Franke. *Programmers Handbook WANDA Server*. Fraunhofer Institute for Production Systems and Design Technology (IPK), Division Security and Testing Technology, Pascalstr. 8-9, 10587 Berlin, Germany, 2003.
20. L.G. Vuurpijl and L.R.B. Schomaker. Multiple-agent architectures for the classification of handwritten text. In *Proc. 6th International Workshop on Frontiers in Handwriting Recognition (IWFHR)*, pages 335--346, Tadjon, Korea, 1998.
21. XSL Transformations (XSLT) - W3C Recommendation. <<http://www.w3.org/TR/xslt>>, 1999.
22. K. Franke, L.R.B. Schomaker, C. Veenhuis, C. Taubenheim, I. Guyon, L.G. Vuurpijl, M. van Erp and G. Zwarts (2003), WANDA: A Generic Framework applied in Forensic Handwriting Analysis and Writer Identification, Design and Application of Hybrid Intelligent Systems, IOS Press, Amsterdam, The Netherlands.
23. M. van Erp, L.G. Vuurpijl, K. Franke and L.R.B. Schomaker (2003), The WANDA Measurement Tool for forensic document examination, Proc. 11th Conference of the International Graphonomics Society (IGS), Scottsdale, Arizona, USA, pp. 282-285.
24. K. Franke, L.R.B. Schomaker, L.G. Vuurpijl and St. Giesler (2003). FISH-new: A common ground for computer-based forensic writer identification. Proc. 3rd European Academy of Forensic Science Triennial Meeting, Istanbul, Turkey, Forensic Science International, Vol.136 Suppl. [Abstract] pp. 84.
25. L.R.B. Schomaker, M. Bulacu and M. van Erp (2003). Sparse-parametric writer identification using heterogeneous feature groups. ICIP'2003: IEEE International Conference on Image Processing, pp. 545-548.
26. M. Bulacu and L.R.B. Schomaker (2003). Writer identification using edge-based directional features. ICDAR'2003: International Conference on Document Analysis and Recognition, pp. 937-941.
27. M. Bulacu and L.R.B. Schomaker (2003). Writer Style from Oriented Edge Fragments. Proc. of the 10th Int. Conference on Computer Analysis of Images and Patterns (CAIP'03), pp. 460-469.
28. L.G. Vuurpijl and L.R.B. Schomaker (1997). Finding structure in diversity: A hierarchical clustering method for the categorization of allographs in handwriting. ICDAR'97 International Conference on Document Analysis and Recognition. Pages 387-393, Ulm.
29. K. Franke, M. Köppen (1999). Towards an universal approach to background removal in images of bank checks - extended version, in *Advances in Handwriting Recognition*, S.-W. Lee (ed.), World Scientific 1999, pp. 91-100.

30. K. Franke, G. Grube (1998). The automatic extraction of pseudo-dynamic information from static images of handwriting based on marked gray value segmentation - extended version, Journal of Forensic Document Examination (JFDE), Volume 11, Fall 1998, pp 17-38.
31. K. Franke, Y-N. Zhang, M. Köppen (2002). Static signature verification employing a Kosko-Neuro-Fuzzy approach, Proc. International Conference on Fuzzy Systems (AFSS), Kolkata, India, pp. 185-190.
32. K. Franke (2001). Region-based signature validation using a Neuro-Fuzzy approach, a study on the application of Soft Computing in handwriting analysis, Proc. 5th International Congress of the Gesellschaft für Forensischen Schriftenuntersuchung (GFS), Bingen, Germany, June 13 - 15, 2001.
33. C. Veenhuis, O. Bünnemeyer, M. Rieß, K. Franke (2001). Automatic screening process for proof of authenticity of German EG stamps under real life conditions, Proc. 5th International Congress of the Gesellschaft für Forensischen Schriftenuntersuchung (GFS), Bingen, Germany.
34. K. Franke and L.R.B. Schomaker (2003), Pen orientation characteristics of on-line handwritten signatures, Proc. 11th Conference of the International Graphonomics Society (IGS), Scottsdale, Arizona, USA, pp. 224-227.
35. K. Franke, L.R.B. Schomaker and W. Penk (2003), On-line pen input and procedures for computer-assisted forensic handwriting examination, Proc. 11th Conference of the International Graphonomics Society (IGS), Scottsdale, Arizona, USA, pp. 295-298.