INF FAKULTÄT FÜR INFORMATIK

Tagungsband der

Magdeburger-Informatik-Tage 3. Doktorandentagung 2014 (MIT 2014)

Herausgeber:

Christian Hansen
Stefan Knoll
Veit Köppen
Georg Krempl
Claudia Krull
Eike Schallehn

8. Juli 2014

Impressum:

Verlag: Otto-von-Guericke-Universität Magdeburg

Verlagsnummer: 85720

Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik
Postfach 41 20
39016 Magdeburg

Herausgeber:

Christian Hansen
Stefan Knoll
Veit Köppen
Georg Krempl
Claudia Krull
Eike Schallehn

Redaktionsschluß: 8. Juli 2014

Redaktion/Gestaltung:
Eike Schallehn

Herstellung:
Magdeburger DigitalDruckerei GmbH

ISBN 978-3-944722-12-2

`http://www.cs.uni-magdeburg.de/`

Vorwort

In diesem Tagungsband werden die Ergebnisse der dritten Magdeburger-Informatik-Tage (MIT) vorgestellt. Diese an Doktoranden der Fakultät für Informatik der Otto-von-Guericke-Universität Magdeburg adressierte Tagung findet 2014 bereits das dritte Mal in Magdeburg statt. In diesem Band sind die wissenschaftlichen Beiträge zusammengefasst, die von ausgewählten jungen Wissenschaftlern der Fakultät zu ihren fortgeschrittenen Promotionsprojekten auf den MIT präsentiert werden.

Diese Vorstellung anerkannter Forschungsergebnisse unserer Fakultät über Fachgebiets- und Universitätsgrenzen hinweg ist eines der Ziele der Tagung. Darüber hinaus werden in jedem Jahr in einem wissenschaftlichen Rahmenprogramm Projekte vorgestellt, und zudem durch eingeladene ehemalige Promovenden der Fakultät Perspektiven für die Fortsetzung der Forschungstätigkeit im industriellen und akademischen Bereich aufgezeigt. Die „Best Contribution MIT 2014“ wird mit einem kleinen Preis geehrt, gesponsert von der Gesellschaft für Informatik, Regionalgruppe Magdeburg.

Der vorliegende Tagungsband dokumentiert sowohl die Vielseitigkeit als auch die Konvergenz der Forschungsaktivitäten von Nachwuchswissenschaftlern an der Fakultät für Informatik. So sind in diesem Jahr erneut Beiträge aus sehr verschiedenen Teilbereichen der Informatik vertreten, die jedoch nicht ganz zufällig einen direkten oder indirekten Bezug zu Sicherheitsaspekten in Softwaresystemen haben. Diese stellen eine wichtige Anforderung an zunehmend vernetzte und eingebettete Computersysteme dar und sind damit von großer Bedeutung für aktuelle Fragestellungen der Informatik.

Das Tagungsprogramm der MIT 2014 am 8. Juli 2014 stellt sich wie folgt dar:

12:45	Eröffnung MIT 2014	
13:00-14:00	Komitee: Ortmeier, Tönnies, Mossakowski, Krätzer, Köppen, Hansen	
13:00	**Eric Clausing** (AG Dittmann)	Digitized Locksmith Forensics: Design and Implementation of a Computer-Aided Forensic Analysis
13:30	**Stefan Kirst** (AG Dittmann)	Digitized Forensics: Segmentation of Fingerprint Traces on Non-Planar Surfaces Using 3D CLSM
14:00-14:30	**Gastvortrag**	**Dr. Andreas Lang**, T-Systems Multimedia Solutions GmbH Dresden
14:30-15:00	Kaffeepause	
15:00-16:00	Komitee: Kruse, Nürnberger, Theisel, Kaiser, Schallehn, Knoll	
15:00	**Andreas Meier** (AG Kruse)	Methods for predicting crash severity prior to vehicle head-on collisions
15:30	**Stefan Haun** (AG Nürnberger)	Exploring the Personal Information Space
16:00-16:15	Beratungspause	
16:15-17:00	**Dr. Henry Herper**, „Digitale Medien in der Lehre“	
	Verleihung des Preises „Best Contribution MIT 2014“ gesponsert von der Gesellschaft für Informatik, Regionalgruppe Sachsen-Anhalt	
Ab 17:00	Verpflegung: Grillen und Getränke vom FaRaFIN, gegen Unkostenbeitrag	

Magdeburg, den 8. Juli 2014

Christian Hansen
Stefan Knoll
Veit Köppen
Georg Krempl
Claudia Krull
Eike Schallehn

Inhaltsverzeichnis

Digitized Locksmith Forensics: Design and Implementation of a Computer-Aided Forensic Analysis

Eric Clausing[a, b]
[a]Otto-von-Guericke University
Magdeburg, Germany
[b]University of Applied Sciences
Brandenburg, Germany
Email: clausing@iti.cs.uni-magdeburg.de

***Abstract*—At the moment, there is a change of paradigm in the field of criminalistic forensics. While the classic approach to forensic analysis, especially in the fields of dactyloscopy and toolmark analysis, heavily and solely relied on experience and training of human experts, the modern forensics more and more advances into a digitized domain. Experts are supported by high resolution sensors, machine learning algorithms and computer visualizations. The goal thereby is not the replacement of human experts but the efficient support to allow for objective, deterministic and reproducible analysis results with less effort, high performance, and known and well documented error rates. For firearm forensics and dactyloscopy there are generally accepted systems existent which allow for a computer-supported, (semi-) automated analysis. Their success motivates the introduction of similar systems to other fields of classic forensics.**

In this paper, we propose the introduction of such a system to the highly specialized field of locksmith forensics. We present a general design of such a system with an exemplary overall analysis goal, discuss challenges, and describe possibilities for an effective implementation. The design covers all necessary steps from acquisition, pre-processing and analysis of a chosen lock component ('key pin'). Our implementation is evaluated with the help of a preliminary test set consisting of a selection of significant samples.

I. Introduction

In modern criminalistic forensic, computer-aided analysis methods get more and more relevant. They provide considerable advantages compared to solely manual analysis methods in respect to performance, objectivity, and reproducibility. Especially in the fields of dactyloscopy (i.e., analysis of latent fingerprints) and firearm forensics there are a quite high number of commercial systems, which allow for a fully- or semi-automated analysis of the particular object of investigation. These systems use high-resolution measurement devices for data acquisition and approaches from the field of machine learning for analysis to allow for detailed interpretations of the found traces and trace complexes.

In this paper, we propose a design and possible implementation of such a system for the field of locksmith forensics as there is none existent at this time. In [1], we firstly propose a complete process model for the digitized forensic locksmith analysis. This proposed model consists of five separate stages which all together form a complete procedure for the determination whether or not a lock has been illegally overridden and if, which opening method has been applied to do so. The proposed five staged model is shown in Figure 1.

The stage of 'Trace Positioning and Acquisition' includes the physical preparation of the object of investigation (i.e., correct alignment under the sensory) and the actual digital acquisition as well as general pre-processing of the acquired data. The goal of Stage 1 'Detection by Segmentation' is the detection and precise masking of all relevant traces. As investigated surfaces in the field of locksmith forensics are often cluttered with traces originating from fabrication (e.g., milling or drilling) it is essential to differentiate between relevant (i.e., not originating from fabrication) and irrelevant (i.e., originating from fabrication) traces to avoid false interpretations. To allow for such a detection, masking, and differentiation, meta knowledge about the characteristic formation and shape of the trace complexes is used to fit standard filtering methods to our special needs. The result of Stage 1 is a binary masking of all relevant trace regions for further investigations. In Stage 2 'Trace Type Determination', we utilize the characteristic shape and texture of the traces of normal wear (i.e., traces originating from normal key usage) to differentiate them from other toolmarks (i.e., traces of a potentially illegal opening attempt). For this purpose, we create a set of features that allows for an automatic classification. By that, we refine our masking by excluding all traces of normal wear as they are not relevant for further investigation. In the third and last stage, the complex formed by all traces that are not originating from wear and fabrication is analyzed, described by a special feature set and classified. The separate stages are described and explained in detail in [1], [2] and [3]. In this paper, we describe how these three stages can be implemented and how they can be connected to form a complete analysis system with an exemplary overall goal. We focus on the lock component 'key pin' as the most reliable and most important region of trace formation. A schematic illustration of a typical pin tumbler locking cylinder (with key pins in black) is shown in Figure 2.

The paper is structured as follows: Following this introduction, we present work relevant for the presented approach. After that, we present details of our concept, challenges met and how the theoretical concept is implemented. Finally we present preliminary results of the automated process, compare them with the results we achieved in former work, where parts of

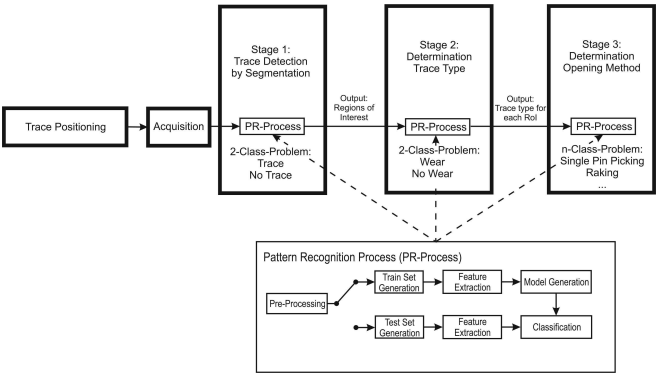


Fig. 1: Process model for digitized locksmith forensics as proposed in [1]

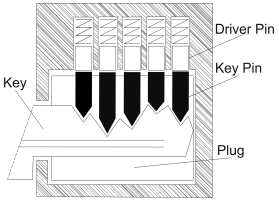


Fig. 2: Schematic design of a standard pin tumbler cylinder lock; key pins shown in black [1]

the process have been performed manually and from that draw important conclusions on how to further improve and evolve the whole process.

II. Related Work

The classic approach to locksmith forensics nowadays is still highly dependent on skill and experience of the examining expert and is only marginally supported by technical measures, namely a classic light microscope. Technical systems which are able to automatically acquire and analyze a toolmark sample are not yet existent for the field of locksmith forensics. Although there actually publications and scientific works in this field like [4][5], none of these consider the automated classification, detection, or even acquisition of such marks. Instead, they mostly address known opening methods and the resulting traces and trace complexes which represents an excellent starting point for the research in this field, but delivers no clue for how to implement an automated analysis.

Especially in regard to classic toolmarks and marks on firearms (projectile and cartridge), there are plenty of scientific papers and technical systems concerning the acquisition, preprocessing, and classification (e.g., [6]). For firearms there are even commercial systems like *IBIS TRAX-3D* [7], which provide a solution for the complete investigation process from acquisition to pre-processing and automated comparison. These systems allow for a reliable identification of a given sample bullet.

In respect to contactless sensors for the acquisition of toolmarks, we are motivated by the *IBIS* system [7], which is using a confocal laser microscope. The high commercial success of the *IBIS* system demonstrates the overall high suitability of the confocal laser scanning approach for purposes quite similar to our own. For our research, we use the confocal 3D laser scanning microscope *Keyence VK-X 105/110* (exact specifications in [8]).

The five stage process model (see Figure 1) this paper bases on is first proposed in [1]. In [1] we additionally propose a solution for the first three steps of the process and discuss the general challenges of transferring classic locksmith forensic to the digital domain. In [2] we propose first approaches for the stages 2 and 3 of the process and again discuss challenges met when implementing and evaluating these stages. Additionally we present a fully automated approach for the pre-processing (especially assembling) of the data acquired according to [1]. In [3] we present an improved version of our segmentation approach of [1] and extend our data pre-processing and assembling to handle topography and color data as well.

A theoretical integration of our proposed process model into the general digitized forensics process is discussed in [9]. There, the theoretical and formal challenges of integrating and implementing such a system are presented and discussed and a general formalization of our proposed process is described.

III. Concept and Implementation

In this section, we describe and explain our overall concept (with all steps according to Figure 1) and the corresponding implementation.

A. Trace Positioning and Acquisition

In this stage the whole raw data acquisition along with first pre-processing steps are performed. To allow for a gapless

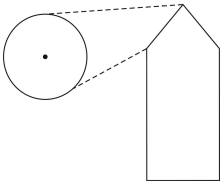


Fig. 3: Schematic design of a standard key pin [1]

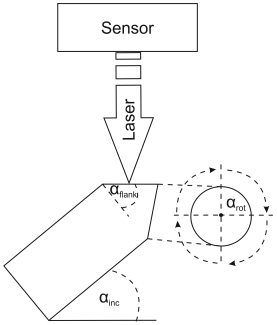


Fig. 4: Proposed trace positioning for a gapless, non-distorted data acquisition [1]

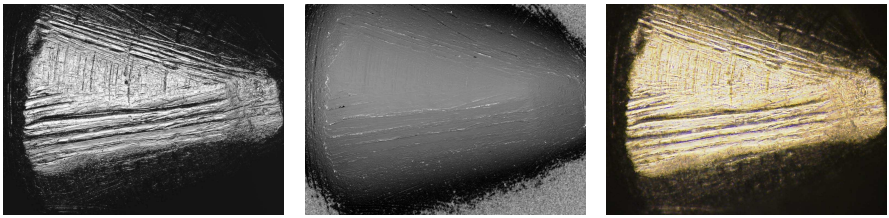
and distortion free acquisition of the key pin surface, an additional positioning of the object of investigation is needed. Although an acquisition of the key pin head from a vertical top view might appear as the most intuitive way, we suggest another positioning. In Figure 3, one can see a key pin from top view. One can clearly recognize and understand the distortion in perspective of the toolmarks on the conical pin head if acquired from top view. Using such acquired data for a detection approach requires a non trivial correction in perspective and an isometric visualization of the 3D cone to a 2D surface - a highly complex task which should be avoided if possible. Instead we suggest to perform this correction before actually scanning the surface by a positioning of the key pin as illustrated in Figure 4 and described in detail in [1]. Essentially it is a parameterizable method to allow for a complete acquisition of the cylindric pin head by consecutively scanning and rotating the pin. The result is a set of partial scans, which all together form a complete representation of the specific surface. Figure 5 shows one partial scan of a key pin surface in all its data representations (intensity, topography, and color).

To allow for an analysis of the whole surface as one, we need to create a complete representation out of the set of partial scans. To do so, we adapt and use the *SIFT*-Algorithm (*Scale Invariant Feature Transform*; [10]) to register and assemble the set of partial scans to one complete representation as described in [2] and expanded in [3]. For this purpose, we apply the algorithm to the set of partial intensity scans, use the so gained information to assemble the color set as well and with some additional processing (due to an additionally necessary alignment in z-direction; more details on that in [3]) the topography data set. The result of the proposed acquisition and assembling of a scan set of 45 partial scans (i.e., acquired with a rotational angle $\alpha_{rot} = 8°$) in all three data representations is shown in Figure 6. This complete representation is used for the actual analysis in the following Stages 1 to 3.

B. Stage 1: Detection by Segmentation

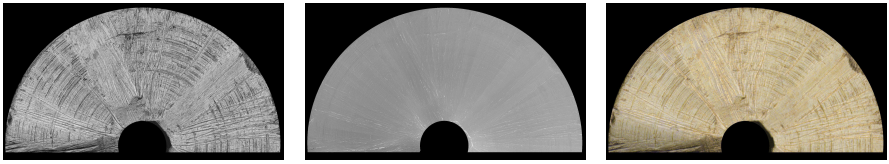
The goal of this stage is the differentiation of toolmarks originating from fabrication and those applied to the surface after fabrication. The second type of toolmarks is the one relevant for further investigation. For an (semi-)automated analysis it is essential to detect and segment these toolmarks as precise as possible. To do so, we use certain meta knowledge about the possible kinds of formation of both types of toolmarks, relevant and irrelevant, to distinguish between both. In [1] we describe how toolmarks originating from fabrication tend to form a pattern of fine circular patterns, whereas toolmarks originating from keys or other tools are more coarse and irregular. The difference between regions with relevant and irrelevant toolmarks is shown in Figure 7. In [1] we use this difference in appearance to perform a texture based differentiation of regions with relevant and regions with irrelevant toolmarks. For texture analysis the Gray-Level-Cooccurrence-Matrix approach (*GLCM*; see [11]) in a blockwise application in combination with a two-class classification is tested in [1]. This approach is based on a blockwise segmentation of the intensity representation of the surface to be investigated, the computation of a set of 160 features (consisting of various statistical features calculated for a number of differently parametrized GLCM's; for more details see [1]) for each block, which is then used to classify between relevant and irrelevant blocks. With this first approach we are able of achieving reliable correct classification rates between 75% and 85% depending on the used classifier.

As the segmentation with a solely blockbased approach is quite coarse and the detection rates are not completely sufficient, we adapt and expand this approach in [3]. For this purpose, we introduce three improvements to the detection by segmentation approach. The first two improvements have the focus on raising the classification rates, the last improvement aims at a refinement of the actual segmentation on pixel level. As the toolmark patterns of the fabrication marks have a different direction depending on the investigated location on the surface (as the data representation created in the acquisition stage is a circle segment; see Figure 6) we introduce a location depending rotational correction of each block to allow for an identical alignment of all blocks (and the potential toolmark pattern) before computing the actual feature set. That has a positive effect on the classification process as the created classification model does not need to consider a possible variable orientation of the toolmark patterns and therefore gets less complex, more performant and more reliable. As second improvement we expand the feature set with additional features extracted from the topography and color data representation of the surface. These additional features include various roughness-based features, further statistical features from color space and additional GLCM computations on topography data (for more



(a) (b) (c)

Fig. 5: Partial scan in (a) Intensity; (b) Topography; (c) Color [3]



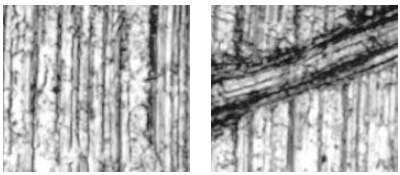
(a) (b) (c)

Fig. 6: Assembled data sets in (a) Intensity; (b) Topography; (c) Color [3]

details see [3]). The total number of features used for classification thereby rises up to 351. The correct classifications are improved up to reliable 90%.

To refine the segmentation precision on pixel level, we combine the texture analysis approach with the Gabor filtering approach (first introduced in [12][13]) to allow for a pixelwise segmentation of regions with potentially relevant toolmark formations. As the Gabor filter can be parametrized to amplify structures in one orientation and dampen other structures in other orientations and as we have meta knowledge about the theoretical orientation of toolmark patterns for a given location on the surface, we want to ignore (i.e., the circular patterns of fabrication), we can use it to amplify everything else (i.e., the toolmarks applied to the surface after fabrication). To perform the amplification of potentially relevant structures, the Gabor filter is also applied blockwisely (although with a different block size as the texture analysis) and in different orientations (which orientations depends on the location of the specific block to be filtered) to the intensity representation. The result are a number of convolutions (one for each filter orientation) for each block which are then combined and transformed into a binary mask. Figure 8 illustrates an exemplary filtering and processing of one block. The specific parametrizations of the approaches (block sizes, Gabor filter parameters etc.) and computations for the determination of the filter orientations are described in detail in [3].

The main problem with the Gabor filtering on surfaces so massively cluttered with toolmarks of all kinds is the high number of false positives. To avoid these false positives and to take advantage of the high reliability of the texture analysis, we fuse the resulting masks of both approaches (more details on the fusion in [3]). The result of the texture analysis and Gabor filtering is shown in Figure 9 and the result of the fusion process is shown in Figure 10.

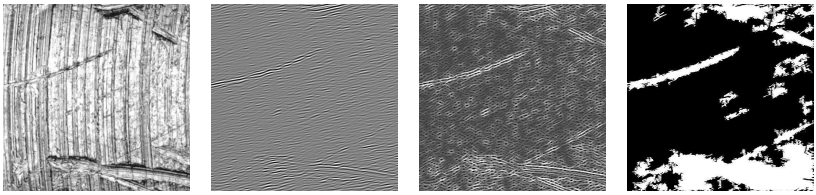


(a) (b)

Fig. 7: Differences between irrelevant and relevant marks; (a) shows a region with only irrelevant marks of fabrication; (b) shows a region with relevant toolmarks (scratches, bumps, abrasion) [1]

C. Stage 2: Trace Type Determination

With the segmentation mask created in Stage 1, we get a binary representation of all regions which contain toolmarks other than those originating from fabrication. At this point this still includes toolmarks from potentially illegal openings as well as toolmarks from normal legitimate key usage. For the goal of determining the most probable opening method applied to the locking cylinder, we have to further differentiate the segmented toolmark regions. In this stage, we model this further differentiation in form of a two-class problem of 'Wear' vs. 'No Wear'. For this purpose, we design a feature set which



(a) (b) (c) (d)

Fig. 8: (a) Unfiltered block; (b) Convolution result of one orientation; (c) Combination of all convolution results in one projection; (d) Segmented trace regions in form of a binary mask (*fill holes* applied) [3]

(a) (b) (c)

Fig. 9: (a) Resulting binary mask from Gabor filtering approach; (b) Resulting binary mask from surface classification; (c) Original assembled intensity data set as comparison [3]

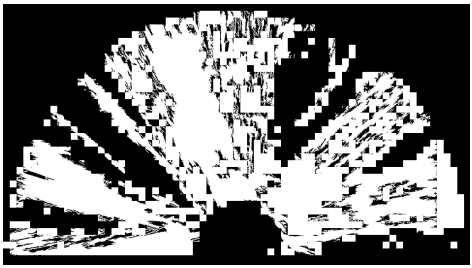


Fig. 10: Resulting mask of Stage 1: 'Trace Detection by Segmentation' [3]

allows the detailed description of a single segmented toolmark region and a classification of all toolmark regions into one or the other class. The feature set, at this point, consists of 17 mainly shape- and dimension-based features (detailed description in [2]).

In [2], we test the approach of a two-class classification of single traces with a test set of over 500 manually segmented traces on the surface of key pins from locking cylinders. The achieved correct classification of traces that are not normal wear is in any case over 95% (see [2]). Of course we have to admit that these results are achieved on manually segmented data, so it is highly probable that the results achieved with an automated segmentation are significantly lower.

D. Stage 3: Determination of Opening Method

This stage represents the overall goal of the whole analysis. The exemplary goal we chose to realize here is actually replaceable with quite a number of theoretically possible goals (e.g., trace age determination or determination of the sequence of application). We chose to perform the determination of the opening method here, because this is most probably the most relevant question after detecting (or assuming) an illegal override of a locking cylinder. In this stage all single traces classified as 'No Wear' are considered in connection and relationship to each other, i.e. the whole trace complex as sum of all relevant single traces is analyzed and classified. For this purpose, a feature set is proposed in [2] that allows for a classification of the potential n-class problem (one class

for every possible opening method). This feature set consists of 20 features based on average, maximum and, minimum single trace features as well as 3 features describing the location and distribution of the traces on the surface (for more details see [2]).
In [2] we test the feature set on a test set of 15 key pins with 15 different trace complexes consisting of about 530 single traces, originating from 3 different opening methods (Wear, Single Pin Picking, Raking). With the proposed feature set and on the used test set we are able of achieving 100% correct classifications with different classifiers. Although we have to admit that the test set is rather small and as for stage 2, the data is manually segmented. With automated segmentation the results are expected to drop.

IV. Test Set and Evaluation

In this section, we present our test concept for an evaluation of our proposed approaches and discuss the achieved results.

A. Test Set

The exemplary test set we use for a preliminary evaluation of our proposed process chain is the same as described in [3]. We use the acquired surface data of 20 key pins from four locking cylinders, all opened with different opening methods. In total the acquired key pin surfaces contain about 700 single traces. The opening methods we applied to the four locking cylinders are *Single Pin Picking* (a high skill lock picking method), *Raking* (a low skill lock picking method), *Pick Gun* (very effective low skill method based on the percussion principle) and *Normal Key Usage* as comparison. The 20 key pins are each acquired in 45 partial scans, which results in a total number of 900 scans, each consisting of three data representations in intensity, topography and color in a resolution of 1024x768. The sensor we use for acquisition is the 3D confocal laser scanning microscope *Keyence VK-X 110* with the following parametrization:

- $mag = 10$, which is the lens magnification
- $stepZ = 0.2\mu m$, which is the z-resolution
- $stepXY = 0.65\mu m$, which is the x-y-resolution
- $\Delta Z = 200\mu m$, which is the z-interval

The resulting 900 partial scans are assembled to 20 key pin head representations as described in Section III-A.
For the Stage 1 segmentation, the parametrization for the Gabor filtering approach is the one determined in wide range testing in [3]. The parametrization with the best error rate ratio for our purpose is the following:

- $size = 512$, which is the block size used for Gabor filtering in pixels
- $\lambda = 4.0$, which is the wavelength of the sinusoidal factor in pixels
- $\psi = 0°$, which is the phase offset of the cosine factor in degrees
- $\sigma = 2.0$, which is the variance of the Gaussian envelope
- $\gamma = 0.5$, which is the spatial aspect ratio of the Gabor functions support
- $\varepsilon = 28°$, which is the considered epsilon environment for α_f in degrees
- $(\theta_1, \ldots, \theta_n)$, which is the tuple of orientations used for the Gabor filtering of a specific block
- α_f is orthogonal to α_o, which is the orientation of the fabrication marks for a specific block

The block size chosen for the surface classification part of stage 1 is 32x32.

B. Results

For the testing of our approach, we perform the whole process as shown in Figure 1 fully automated to each of the 20 key pin representations. We use a balanced set of different classifiers to exclude the possibility of an overfitting of our approach for a special classifier. For classification, we use the *WEKA* data mining software in version 3.7[1]. All classifiers are used in their standard parametrization and originate from the following classes of classifiers (specifically used classifiers are written in brackets): Bayes (*Naive Bayes*), functional (*RBF Network*, *SMO*, *Simple Logistic*), Lazy (*IB1*), Meta (*Bagging*, *Random Committee*, *Random Subspace*, *Rotation Forest*), Rule-based (*Decision Table*, *OneR*) and Tree-based (*J48*, *Random Forest*, *Random Tree*). Due to the quite small number of instances, we use a full 10-fold cross validation for the determination of the performance rates.

1) Stage 1: Detection by Segmentation

To validate the results, we chose to measure the overall performance of our Stage 1 approach with standard performance criteria. These performance rates are:

- *True Positive Rate* (TP), which is the percentage of pixels/blocks correctly segmented as part of a trace not originating from fabrication.
- *True Negative Rate* (TN), which is the percentage of pixels/blocks correctly recognized as part of a fabrication mark.

For the Gabor filtering part, we use our approach on every instance of the 20 assembled intensity sets and calculate the average performance by comparing with manually segmented reference masks. The detailed resulting values our approach is able to achieve for stage 1 are presented [3]. For the block-based classification with the presented surface feature set, we are able of achieving reliable TP rates of over 90% combined with quite high TN rates of over 80%. Especially the tree based classifiers, as e.g. *Random Forest* and *Rotation Forest*, seem to be well fit for our purpose as they provide the best TP/TN combinations for all opening methods. For the Gabor filtering alone, we achieve a top TP of about 70% and a maximum TN value of 81% (both for the opening method *Raking*). The fusion is able of achieving TP values of about 90% and TN of about 80%. It is noticeable that in all cases the TN values for the fusion are significantly better than for surface feature classification and Gabor filtering alone although the TP rates slightly decrease on pixel level.

2) Stage 2: Trace Type Determination

To evaluate the overall performance of our Stage 2 approach, we chose as well standard performance criteria. These performance rates are:

[1]Weka 3: Data Mining Software in Java; http://www.cs.waikato.ac.nz/ml/weka/; Version 3.7

	TP	TN	Kappa
Naive Bayes	0.99	0.67	0.50
Simple Logistic	0.99	1.00	0.86
SMO	0.995	0.00	0.0
IB1	0.995	0.67	0.57
Bagging	1.00	0.00	0.00
Random Committee	0.99	0.33	0.40
Random Subspace	1.00	0.00	0.00
Rotation Forest	1.00	0.00	0.0
J48	0.99	0.00	0.00
FT	0.99	0.67	0.66
Decision table	0.99	0.33	0.40
OneR	1.00	0.67	0.80
RBF Network	0.99	0.67	0.57
Random Forest	0.99	0.67	0.66
Random Tree	0.99	0.67	0.57

TABLE I: Results for Stage 2: 'Trace Type Determination'

	Acc_{Wear}	Acc_{Raking}	Acc_{SPP}	Acc_{Pick}	$Acc_{Overall}$	Kappa
Naive Bayes	1.00	0.67	0.33	0.33	0.58	0.44
Simple Logistic	1.00	0.00	0.67	1.00	0.67	0.56
SMO	1.00	0.67	0.67	0.67	0.75	0.67
IB1	1.00	0.67	0.67	1.00	0.83	0.78
Bagging	1.00	0.00	0.33	0.00	0.33	0.1
Random Committee	1.00	0.33	0.33	0.67	0.58	0.44
Random Subspace	0.33	0.00	0.00	0.00	0.08	-0.22
Rotation Forest	1.00	0.33	0.67	0.67	0.67	0.56
J48	1.00	0.67	0.33	0.33	0.58	0.44
FT	0.00	0.00	0.00	0.00	0.00	-0.33
Decision table	0.67	0.00	0.33	0.00	0.25	0.30
OneR	0.00	0.00	0.00	0.00	0.00	-0.33
RBF Network	1.00	0.67	0.33	0.00	0.50	0.33
Random Forest	1.00	0.00	0.67	0.33	0.50	0.33
Random Tree	0.67	0.33	0.00	0.33	0.33	0.11

TABLE II: Results for Stage 3: 'Determination Opening Method'

- *True Positive Rate* (TP), which is the percentage of single traces correctly classified as 'No Wear'.
- *True Negative Rate* (TN), which is the percentage of single traces correctly classified as 'Wear'.
- *Kappa statistic* ($Kappa$), which measures the agreement of prediction with the true class. It is a value between -1.0 and 1.0, where 1.0 signifies complete agreement with the true class, -1.0 is the inverse of the complete agreement and 0 is basically guessing. We use this additional measure to compensate for the fact that the number of instances in each class is not equal and therefore influences the results of the TP/TN calculations.

As can be seen in table I, the results for Stage 2 on automatically segmented data are quite comparable to the results presented in [2] (where tests are performed on manually segmented data). However, the number of instances here in this work is about half of the size of the one used in [2]. Although a larger set of key pins is used, the automated segmentation approach tends to deliver greater regions with more than one single trace in it. Consequently the total number of regions and thereby the total number of instances for testing drops with the automated segmentation. However, for the smaller test set the automated segmentation does not seem to negatively affect correct classification rates. In case of the classifier *Simple Logistics* we can actually notice a significant improvement with a TP of 99% and a TN of 100% with a really good kappa value of 0.86. The reasons for the outstanding performance of exactly this classifier for our purpose has to be topic of further investigation.

3) *Stage 3: Determination Opening Method*

To evaluate the overall performance of our Stage 3 approach (and thereby the performance of the whole system), we chose as well standard performance criteria for classification problems with multiple classes. These performance rates are:

- *Accuracy for classification of 'Wear'* (Acc_{Wear}), which is the percentage of instances correctly classified as 'Wear'.
- *Accuracy for classification of 'Raking'* (Acc_{Raking}), which is the percentage of instances correctly classified as 'Raking'.
- *Accuracy for classification of 'SPP'* (Acc_{SPP}), which is the percentage of instances correctly classified as 'SPP'.
- *Accuracy for classification of 'Pick Gun'* (Acc_{Pick}), which is the percentage of instances correctly classified as 'Pick Gun'.
- *Overall Accuracy* ($Acc_{Overall}$), which is the overall percentage of all correctly classified instances.
- *Kappa statistic* ($Kappa$), which is the Kappa statistic for the whole four-class problem.

For Stage 3 the classification results are shown in Table II. In comparison to [2], the results are altogether significantly lower with a top value for the classifier *SMO* with an overall accuracy of 75% with a kappa value of 0.67. We assume two facts to be responsible for this. Firstly, the data set here is expanded with an additional opening method (Pick Gun), which leads to a four-class problem instead of the three-class problem in [2] and thereby a more complex classification model. Secondly, our automated segmentation approach in Stage 1 tends to comprehend multiple single traces to on segmented region. However, the feature set for Stage 3 heavily relies on averaged trace shape and dimension of all traces in one trace complex. This information is significantly corrupted when it is computed for whole trace regions instead of each single trace separately.

V. Conclusion and Future Work

In this paper, we present a design and possible implementation of a forensic analysis system for locksmith forensics. The implementation allows for an automated analysis of key pins with the analysis goal of determining whether the components of a lock show any signs of a possible illegal opening and if so, which opening method has most probably been applied to do so. For this purpose, we provide a five step process with an evaluated implementation for each of the two acquisition steps ('Trace Positioning' and 'Acquisition') and the three analysis stages. The results, we are able to achieve for all stages, are mostly good and in any case demonstrate the general feasibility of such a system for a locksmith forensic analysis. Although we are able of achieving quite good results, there is plenty of room for improvement in each of the stages. Especially

in terms of a further refinement of Stage 1 and a feature set expansion for Stage 2 and 3 seem to be the best ways to significantly improve our proposed process. For Stage 1, the goal must be the separation of single traces within the segmented regions, to allow for a more precise description of these traces for Stage 2 and 3. For Stage 2 and 3 themselves, we plan to expand the feature set with additional features from the data representations topography and color. We are confident of extracting additional useful information from these data representations to allow for a more precise trace and trace complex description and thereby better classification results.

ACKNOWLEDGMENT

The work in this paper has been funded in part by the German Federal Ministry of Education and Science (BMBF) through the Research Program under Contract No. FKZ:13N10818 and FKZ:13N10816. Furthermore the authors want to thank Jana Dittmann (Otto-von-Guericke University Magdeburg) and Claus Vielhauer (Brandenburg University of Applied Sciences) for many fruitful discussions and valuable suggestions.

REFERENCES

[1] E. Clausing, C. Kraetzer, J. Dittmann, and C. Vielhauer, "A First Approach for the Contactless Acquisition and Automated Detection of Toolmarks on Pins of Locking Cylinders Using 3D Confocal Microscopy," in *Proceedings of the on Multimedia and security*, ser. MM&Sec '12. New York, NY, USA: ACM, 2012, pp. 47–56.

[2] ——, "A first approach for digital representation and automated classification of toolmarks on locking cylinders using confocal laser microscopy," in *Proc. SPIE 8546, Optics and Photonics for Counterterrorism, Crime Fighting, and Defence VIII, 854609*. SPIE, September 24-27 2012.

[3] E. Clausing and C. Vielhauer, "Digitized locksmith forensics: automated detection and segmentation of toolmarks on highly structured surfaces," in *Proc. SPIE Media Watermarking, Security, and Forensics 2014*, vol. 9028, February 19 2014, pp. 90 280W–90 280W–13.

[4] DATAGRAM, "Lock Picking Forensics," [Online] available: http://www.lockpickingforensics.com, last checked 05/05/2014, 2014.

[5] ——, "Lock Wiki," [Online] available: http://www.lockwiki.com, last checked 05/05/2014, 2014.

[6] D. Li, "Ballistics Projectile Image Analysis for Firearm Identification," in *IEEE Transactions on Image Processing*, vol. 15, 2006, pp. 2857–2864.

[7] Forensic Technology Inc., "The Development of IBIS-TRAX 3D: BulletTRAX-3D and BrassTRAX-3D," [Online] available: http://www.forensictechnology.com, last checked 05/05/2014, 2014.

[8] Keyence Corporation, "VK-X100/X200 Series 3D Laser Scanning Microscope," [Online] available: www.keyence.com/products/microscope/microscope/vkx100_200/vkx100_200_specifications_1.php, last checked 12/17/2013, 2013.

[9] S. Kiltz, E. Clausing, J. Dittmann, and C. Vielhauer, "Ein Vorgehensmodell für die digitale Schlossforensik," in *D-A-CH Security 2013 - Bestandsaufnahme, Konzepte, Anwendungen, Perspektiven*. IT-Verlag Sauerlach, 2013, pp. 367–379.

[10] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," in *International Journal of Computer Vision*, vol. 60 (2), 2004, pp. 91–110.

[11] R. Haralick, K. Shanmugam, and I.Dinstein, "Textural features for image classification," in *IEEE Transactions on Systems, Man, and Cybernetics SMC*, vol. 3 (6), 1973, pp. 610–621.

[12] J. Daugman, "Uncertainty relations for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," in *Journal of the Optical Society of America*, vol. 2, 1985, pp. 1160–1169.

[13] ——, "Complete Discrete 2-D Gabor Transforms by Neural Networks for Image Analysis and Compression," in *IEEE Trans on Acoustics, Speech, and Signal Processing*, vol. 36 (7), 1988, pp. 1169–1179.

Exploring the Personal Information Space

Stefan Haun
Data and Knowledge Engineering Group Faculty of Computer Science
Otto-von-Guericke-University Magdeburg, Germany
stefan.haun@ovgu.de
Supervisor: Prof. Dr.-Ing. Andreas Nürnberger

***Abstract*—Advances in digital personal management, such as smart phones, tablets and cloud services, have lead to a massive amount of digitized personal information, but also a quite complete coverage of the personal live with digital artefacts. However, tools for handling different types of information are still separated, making it hard to develop applications that operate on the whole set of personal information. The goal of the thesis is an integrated view on the Personal Information Space, containing all digitised personal artefacts, that allows exploration and advanced operations without complex software integration processes and overcoming the enclosing behaviour of Semantic Desktop solutions. Reaching the goal means that software tools can use any type and part of personal information without having to care for storage and retrieval specialities, finally leading to a richer tool set and better experience for the user.**

I. Introduction

With the advent of ubiquitous electronic devices, such as smart phones, tablets and notebooks, most information is created, stored and manipulated in digital form. As a result there is an almost complete coverage of personal information, like documents, messages or contacts, that is available for digital processing. Personal Information Management (PIM) can be done paperless. However, each type information and its respective tools are still separated, even when stored and run on the same device. *Semantic Desktops* tried to overcome the separation, but did not turn out to be the next-generation information management tool. They are monolithic and do not play well with external tools, such as an additional software solutions or a smart phone accessing the same e-mail account. This leads to a vendor lock-in in terms of information management, as changes are often stored in local databases and would be lost if the user decided to abandon the Semantic Desktop.

The integration of different types of information and tools is complex for several reasons: Most software tools are monolithic and proprietary. They are tailored towards a certain information type, for example e-mails or appointments, and contained their own facilities for storage and processing. Integrating different information types requires integration of different software solutions from different projects, which either requires a common interface or a common development process between those tools. This becomes even harder if more than two information types are involved. The past has shown, that coordination between independent software projects has its caveats and will lead to problems in the long run.

To solve the barrier between different types of personal information, an integrated view is needed. The view would allow easy access to all personal digital artefacts. It would not lock out any tools, even on external devices. And it must be tailored towards the specific needs of personal information, especially an immediate propagation of changes. It is not acceptable for a user to add an appointment on the smart phone, but to see this change on the computer not before some hours later.

Having this integrated view enables further advances: Additionally to navigation and look-up, the user can explore the whole data set to find previously unknown connections. Advanced operations include analysis such as pattern mining over the complete set of personal information, rule-based semi-automated reactions to events such as new messages or incoming appointments, or agent support such as showing additional information regarding the current task at hand. Tools can be developed to operate on the personal information space without having to worry about how and where information is stored and retrieved. Novel developments such as *Google Glass* can profit from an integrated view to enable new applications, like overlay information based on face recognition.

The goals of the thesis are

- to develop a concept for an integrated view on the Personal Information Space
- to prove the concept with a prototype implementation
- to show that novel PIM applications can be enabled by the integrated view
- to provide Graph Exploration as additional search paradigm on personal data

This paper is structured as follows: After a review of Related Work, the Personal Information Space is defined, proceeded by a description of the Integrated View and Exploration as an interaction paradigm. In Advanced Operations the benefits of the thesis results are elaborated, followed by a description of the Validation and finished by a Conclusion.

II. Related Work

In the past, several *Semantic Desktop* solutions tried to unify personal information and provide the user with a common interface to all information types. The *NEPOMUK* project[1], former EU project, contributed towards common information handling, e.g. with ontologies for each type of personal information artefact. *OpenIRIS*[2] offered an open-source solution for a semantic desktop, however suffered the lock-in problem and, as of today, seems to be discontinued. With a slightly different direction, *DeepaMehta*[3] advertises as a software platform for

[1] http://nepomuk.semanticdesktop.org/
[2] http://openiris.org/site/home
[3] http://www.deepamehta.de/

knowledge workers. All information items belonging to a work context are represented in a graph view and can be navigated and manipulated. However, it does not offer an integrated view on the included data sources.

DeepaMehta represents the current trend towards a more open environment. Instead of complete semantic desktops, that try to replace the traditional desktop environment, modern tools integrate data sources and offer specific interactions on those sources. *Everything Is Connected*[4] allows the user to specify a person of history and a location, i.e. a city, to tell a story of how those two items are connected, using the DBpedia—a semantically enriched version of the Wikipedia—as an information source [21]. The *Linked Data* initiative [5] uses the Web to connect previously unconnected data sets on the base of semantic web technologies [4]. In a way Linked Data is the global, static solution of the view that shall be achieved for the personal information space. The KDE project[6] tries to build a semantically extended desktop, thus offering a smooth transition without lock-out effects. Much contributions from the aforementioned NEPOMUK project went here. In general, modern systems come with *content providers*[7] that offer an abstraction layer around storage locations and are the first step towards an integrated view. *Mediator* systems [14, 19] remedy the lock-in problem of semantic desktops: While a data warehouse materialises the integrated view into an internal store, mediator systems keep data in the original sources and distribute the queries.

In contrast to the short request-response process in an (often keyword-based) ad-hoc search, *exploration* is an ongoing process that enables the user to learn about the information space and refine the information need until the desired piece of information has been acquired [15]. While there are many graph layouting methods [6, 12, 13], most of them rely on a static graph structure. In exploration, however, the graph is gradually changed over time, while the user expects a layout with stable node locations [17]. Currently, there are no known graph layouts that fulfil this requirement.

The research topic has to be distinguished from *E-Discovery*, which has its origin in research for legal cases. There, a fixed data set is used, so that applied methods and their results will be valid for the remainder of the (re-)search process.

III. Personal Information Space

Boardman [5] provides three approaches towards a specification of information in Personal Information Management (PIM):

1) information about an individual, e.g. information stored by an institution about an individual
2) information managed and stored within personal organiser software
3) information owned by an individual, and under their direct control

[4]http://everythingisconnected.be/ (works best with *Chromium*)
[5]http://linkeddata.org/
[6]http://www.kde.org/
[7]c.f. the Android content provider: http://developer.android.com/guide/topics/providers/content-providers.html

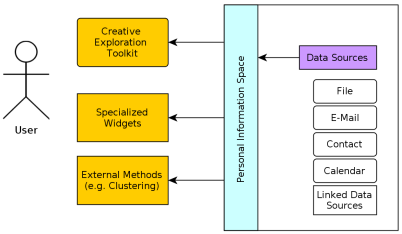


Fig. 1: Structure of the Personal Information Space

Keeping the user's perspective in mind, the third definition applies to personal information in the sense of a Personal Information Space.

The Personal Information Space (see Fig. 1) is a transparently integrated view on all personal information available to the user on his systems. Ideally, this view is identical to the user's mental model of his personal information or even more complete, taking into account that a user will not keep the whole information set in mind.

Two main requirements must be met:

1) Personal information is very volatile and changes as the user interacts with his environment. Those changes must be visible immediately so that the user has a real-time view on his data, instead of appearing some hours late because database updates are done only twice a day.
2) As it becomes more and more common to use more than one device, for example a notebook *and* a smart phone, personal information must be edited *in place*. Otherwise the solution is monolithic in the sense that changes to the personal information is not visible to outside applications. For example, if a user reads an e-mail on the smart phone, this e-mail must be instantly marked as read in the integrated view on his notebook.

In order to represent the Personal Information Space, the *Resource Description Framework (RDF)*[8] is used. RDF is a data model for linked entities with semantic annotation that allows generic reasoning and software-automated processing on arbitrary information resources. A basis for processing are *ontologies*, which define entity types (classes) and relationships between classes. During the *NEPOMUK* project, an ontology for personal information management, *PIM-O*[9], has been developed, which will be used in the integrated view on the Personal Information Space. Additionally, there is the *Friend-of-a-Friend (FoaF)*[10] ontology which only covers contacts, but is very widespread and therefore will be supported.

For the personal information space, the following research questions arise:

[8]http://www.w3.org/RDF/
[9]http://dev.nepomuk.semanticdesktop.org/wiki/PimoOntology
[10]http://www.foaf-project.org/

- Can all relevant PIM data sources be abstracted into the Personal Information Space?
- What is the optimal set of ontologies?
- Are there conflicts between ontologies, e.g. PIM-O and FoaF, and how can they be resolved?

IV. Building an Integrated View

A. Integrated View

Semantic Desktops (SD) often integrate personal information by building a local data warehouse. This means that based on a schema integration process, each entity is copied from the source to the internal database, including necessary changes or addition for the integration layer. As a result of this materialisation process, methods from the SD can now operate on the integrated view. However, changes will not be visible to software outside the SD. Since the data was copied into an internal database, changes to the original source would not be visible within the SD either. This can be solved by running updates from external sources, leading to two additional problems:

1) It takes time until outside changes are visible within the SD. Personal information can be very volatile and changes must be reflected by the integrated view immediately.
2) If both databases are changed simultaneously, it is very to resolve conflicts. The user may not even remember which version is the right one and data becomes unreliable.

Even when data is kept in its original source, indexes for faster access or additional linking add data that will be invisible for outside applications or lost if the user chooses to abandon the SD.

As a solution, a Mediator is used instead of a Data Warehouse. Now, instead of collecting data into an internal database, queries are distributed to the external sources and the result is integrated into the view. Updates to the external sources will be visible instantly and any change to the view is committed to the external sources, i.e. there is only one storage location which cannot be outdated or result in conflicts. Yet the Personal Information Space can be accessed via the integrated view as well as via the original data sources.

Normally, schema integration by a database developer would lead to the schema of an integrated view. However, in the Personal Information Space many different sources must be integrated independent from each other in a generic way. Adding another source, e.g. from the linked data set, must not influence other data sources or their integration. The Resource Description Framework (see Section III) allows to annotate each entity with its semantic meaning. Through lifting [1], external data sources are converted to a RDF representation and virtually added to the integrated view, which by concept is a large graph of semantic entities and their relationships.

B. Operations

Operations on the integrated view can be parted into *read* and *update* operations: Reading from the view includes *Look-Up* of a specific entity based on a key and *Navigation*, which allows to follow a path of linked entities, i.e. a file path through a list of folders. An additional search paradigm is provided by *Exploration* (see Section V). However, integrated database systems normally do not offer updates. It is one of the research questions how the update operations *Add*, *Delete* and *Update* can be implemented based on the information from the integrated view.

For accumulated values or items, which have been assembled from multiple sources, the update operations may not be as straightforward as listed above. Not every data source may support the desired update. Accumulated values often can only be changed by manipulating the base data, e.g. a set of entities. The mediator has to keep track of which interactions are available on the presented data. Further research will focus on the following propositions, both having their advantages and drawbacks:

1) On each intended update the system can perform a dry run and report whether the operation would be successful. While this solution is relatively easy to implement, it is not acceptable from a user's point of view. Neither is the system able to tell whether a planned operation would be possible, nor is it possible to enumerate available interactions to be presented to the user. These limitations deter the user from building a successful mental simulation towards the solution of a task at hand, therefore make it very difficult to achieve a specific goal when interacting with the system.
2) Based on limitations stated in meta-information about data sources, the mediator can keep track of constraints towards the interactions available for the system. From those constraints a set of interactions can be derived for each item in the graph and be presented to the user. This solution, however, results in a much higher effort on developing, implementing and running the mediator system.

As there may be data sources with similar semantics, e.g. person profiles from social networks, it may not always be possible to decide which data source should be changed in order to achieve a certain state. This especially applies to the *Add* operation, as there is no history or meta-data for a newly created entity which would allow to map it to a data source. A disambiguation process can be implemented in several stages:

1) The user is presented with a list of possible actions through the user interfaces, from where he is asked to select one to his like. This solution has two major drawbacks: First, there must be a user interface at all, which might not necessarily be the case with agent-based systems. Second, the user might not know or might not want to be concerned with the selection of an appropriate data source to be changed. This form of presentation breaks the unified view on all data sources.
2) There is a reasoning mechanism which allows to determine the best action to be taken. This might be achieved by a ranking of all possible changes, based on meta-information about the data sources provided by their wrappers. This ranking, however, will be very closely tied to the actual application and

must be carefully designed to reflect the user's needs, otherwise odd decisions may lead to confusion. Still, there is a Semantic Gap between a user's interaction and his intent, for example an application could not easily deduce in which contact to store a just added telephone number. Unless there are clear directions about where to put specific data, the user may still need to make the decision.

3) The reasoning may be supported by finding similar data and deducing the target data source by these elements. This approach is based on the assumption that a user intends to keep the principal structure of his data models. So when a telephone number is added to a contact, the system tries to determine the source which is most likely to contain telephone numbers and puts the number there. Previous choices by the user may be incorporated.

C. Identifiers

In order to link or reference entities, persistent identifiers are needed. *Uniform Resource Identifiers (URIs)* [2] offer a standardised solution towards entity identification. However, in the context of identification two research questions arise:

- How can broken links be avoided by URI scheme design?
- How to recover from a broken link if it occurs nonetheless?

Stability is a key feature of identifiers in the integrated view and since there is no internal database, it must be derived from the data source itself. Unfortunately, this is not reflected in many standards' definitions. For example, the IMAP URI[11] scheme [16] uses the path to an e-mail for identification. This link is broken as soon as the containing folder is re-ordered or the message is moved to another location. Both are common operations in IMAP stores. E-Mails, on the other hand, provide a `Message-ID` field for reference to a particular version of a particular message [18]. This ID is by convention globally unique[12]. On the basis of a stable identifier, an internal index (Figure 2) may be used to increase the performance of a look-up process, if the IMAP source does not support an efficient search by message ID. Since the identifiers are also applicable to the original source, no information is lost if the index is dropped. Files in the *Personal File System* can be identified by their path [3]. However, if a file is moved, the path reference is broken. Using a hash value, as implemented in *magnet links*[13], allows to retrieve a file without knowing the path, as long as the content is not changed (see Fig. 3). When a file is moved and changed, further heuristics must be applied to recover its location, e.g. URIs which are augmented with data from index vectors to recognise the file's content [20].

[11] Based on the notion that a location must always be resolved from an identifier it is nowadays common to use URI as well for URL (Uniform Resource Locator). The URL is a special case where the location is already contained in the identifier. Even though the term URL may appear in references, only URI will be used in this paper.

[12] Although message IDs may be spoofed and are not controlled, Mail Transfer Agent implementations do their best to avoid any clashes with already existing IDs.

[13] http://magnet-uri.sourceforge.net/

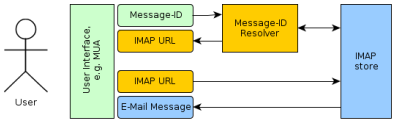


Fig. 2: Message-ID resolver backed by an IMAP store

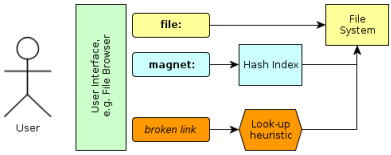


Fig. 3: File resolver for a Personal File System

D. Communication

The complex and diverse setup scenarios call for a flexible communication framework which adds an abstraction layer that allows to communicate with specific peers without knowledge of there whereabouts or technical communication channels. Additionally, communication schemes more capable as the traditional request-response paradigm are needed, especially in the user interface where intermediate results and progress information may be displayed.

The *GLUE* library[14] simplifies communication between heterogeneous software components. It supports various exchangeable transport protocols, so that data can be easily transmitted in different settings: in-memory within a single *Java Virtual Machine (JVM)*; over the wire (IP socket); or even using a chat-like protocol (XMPP[15]). GLUE provides a communication channel which is agnostic of the actual transport method and thus allows a flexible wiring of components.

On top of GLUE lies the *MOCCA* library[16] as a Message-Oriented Command and Context Architecture, providing a middle-ware that allows sending commands to a peer, which are executed by state-less handlers in a peer-specific context. This context can be used to store and access data and will be provided with every call of those handlers for effortless state modelling. In contrast to the request-response paradigm the message flow is not fixed by the framework. This allows the implementation of additional communication schemes. The whole system can be seen as an automaton with Messages that trigger state transitions in the local Contexts.

Two communication settings are used: in-memory communication for components running in the same JVM, and XMPP communication for components on different machines or in different processes.

[14] https://projects.dke-research.de/redmine/projects/glue/

[15] Extensible Messaging and Presence Protocol, http://xmpp.org/

[16] https://projects.dke-research.de/redmine/projects/mocca/

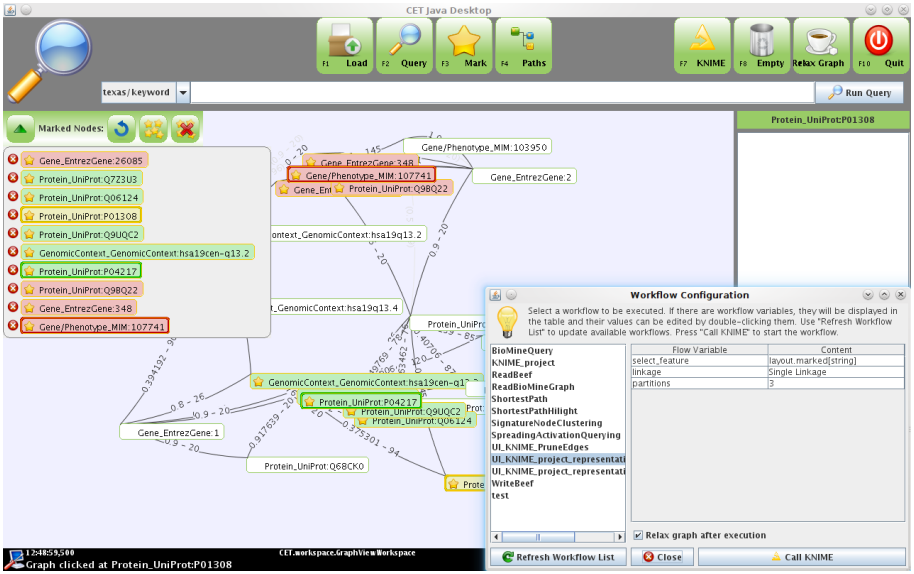


Fig. 4: Screenshot of the Creative Exploration Toolkit (CET), showing an exploration sub-graph from a Gene database and two marked clusters.

V. Exploration

A. Paradigm

Currently, *search*—be it in the World Wide Web or on the desktop—often means ad-hoc keyword search. Based on one or more keywords entered by the user, a result list is generated and presented. If the desired is not in the list, the user starts over and enters a refined or different query term. Ad-hoc search is good for closed fact queries, when the answer can be easily expressed in a query and the user knows if the result is complete. Looking up the local temperature is a typical ad-hoc query task, while finding a suitable restaurant often is not.

Exploration offers a different search paradigm. Instead of a one-shot ad-hoc search, the user is guided through an iterative process that eventually leads to a relevant result. Exploration starts with a pivot element, which may be retrieved by arbitrary means, such as an ad-hoc search, entities from a collection or a recommender. From the result, the user selects one or more elements that fit his information need best. Based on these items, the result is expanded or adapted and the user is queried for another item set. This way the user chooses the direction of each expansion. Due to the iterative nature of the process, even the information need may shift during each step as the information space becomes more and more visible. The user decides when the result is sufficient and the process is finished.

In *graph exploration*, the user selects from a set of nodes in the sub-graph that is currently presented. Each expansion adds the nodes' neighbours and the corresponding links. Since the structure of the information space is shown, the user may discover unexpected links or relationships, for which he would never have searched.

B. Creative Exploration Toolkit

The *Creative Exploration Toolkit (CET)*, shown in Figure 4, has been developed to facilitate graph exploration on generic graph sets. It allows interaction with a dynamic graph derived from an information network, and has an internal graph representation which is agnostic towards the actual domain of the graph's content, leading to a tool that can be used for exploration in various types of explicitly linked graphs. Instead of integrating all methods into on runtime, the aspects *data access*, *layout calculation* and *graph presentation* are separated. During the exploration process, the User Interface initiates the expansion of a pivot element. A graph interaction agent queries the data source for completion of the information space around the pivot element, i.e. nodes in the semantic graph directly connected to the pivot element, and then calculates new positions regarding the graph stored in the context and communicated to the UI. Afterwards the UI displays new nodes and updated node positions. As graph and layouting information are stored in a backend, a client can be rather small, such as a Web-based or smart phone client. In the actual

application context the components can be distributed over several computation devices or put together into one standalone runtime environment.

Graph layouting is a challenge in the exploration context: While graph visualisation is well researched, most established methods rely on a graph being stable. Small changes in the graph topology, such as adding a node, often results in large changes to the layout. As the human brain is especially capable of remembering the location of things [17], it is a requirement for graph exploration that even for a changing graph all node positions must be relatively stable. A variant of the *Stress Minimization Layout* [13] is used to determine the initial graph layout, followed by an overlap removal [6]. Pre-established node positions are taken into account, although they do not generally overrule the layouting process. As a result the calculated graph layout may not be optimal regarding the input graph structure, but is much better suited for interaction in a dynamic environment.

C. Complex Nodes

Complex nodes are an envisioned extension to the CET. Most PIM concepts are too complex to be described by just one node. For example, an *E-Mail* specification in the *NEPOMUK Message Ontology*[17] defines several sub-classes that contain meta-information, such as the recipients. In terms of usability, however, it is better to represent each e-mail as a single node and display the meta-information by means of the node representation, so that the user can easily recognise the documents. Adding each sub-node only occludes the graph structure and makes it harder to understand.

To realise complex nodes, the following questions must be answered:

- How can complex nodes be identified in the RDF graph? This requires to find the best subsumption scheme and resolve ambiguous structures. It might be necessary to allow multiple assignments for a node, i.e. a *Person* taking part in several *E-Mails*, and lead to artificial edges.
- How can exploration queries be built from complex nodes? Three naive solutions come to mind: use the topmost node's URI, query for all URIs in the complex node, or manually select the pivot form the complex node (see concepts like the *Semantic Flower*). However, it would be more interesting to take the semantics of the concept represented by the complex node into account in order to derive the optimal query.

VI. Advanced Operations

Having an integrated view on the Personal Information Space enables numerous new applications that rely on easy and complete information access.

A lot of application examples seem trivial, but they take up a relevant amount of time in everyday tasks:

- On storing a file, the matching target folder is already proposed.
- On opening the e-mail client, the person to reach is selected.
- On receiving new e-mails, immediate notifications are only sent for those e-mails that are relevant for the task at hand, keeping the number of interruptions low without missing urgent information.
- An appointment is coming close, relevant action items and documents pop up with time to spare for their completion.

When developing an application, much time is spent on information storage and retrieval. This not only binds development resources, but also leads to inflexibility: The developer has to foresee all deployment environments. With an integrated view, the application just uses information without caring about data sources or their handling.

Looking at ongoing developments, the integrated view can be applied to new technologies:

- Using head-up displays, such as *Google Glass*[18] or similar technologies, conversation peers can be identified using face recognition. Relevant information regarding the contact will then be visible to the user, reminding him of important tasks and information.
- Using *Near Field Communication (NFC)* or *SmartCard*-based identification, additional information can be used by *Companion Systems*[19] to enrich interaction with available personal information.
- With data mining methods, common behavioural patterns could be discovered and reported to the user, leading to a more efficient task solving or allowing him to become aware of typical processes.

Devices such as Google Glass may not be capable of processing the integration themselves, while data mining applications are hindered by the fact that the integration has to be implemented first. With an integrated view the necessary information space is readily available and can just be used.

VII. Validation

To show that the Personal Information Space can be defined, the relevant PIM data sources must be identified and it must be shown that those sources can be abstracted into an integrated view. As a result, there will be a list of relevant ontologies and a survey showing that this list is sufficient for personal information management. If there are conflicts between the ontologies, a method for conflict resolution will be provided.

Evaluation of the integrated view will be done by setup of a test environment containing relevant PIM data sources. The criteria are: Can all information be accessed? Can changes be propagated in both directions? Can unambiguous queries be resolved? The defined operations for the integrated view can be tested in this setup. Additionally, a sub-set of advanced operations will be implemented to prove the working of the Personal Information Space. A challenge in validation is to

[17] http://www.semanticdesktop.org/ontologies/2007/03/22/nmo/#Email

[18] http://www.google.com/glass/start/

[19] see for example http://www.sfb-trr-62.de/

find a representative data set without violating privacy. For e-mails, the *ENRON Dataset*[20] is a useful resource.

While the efficiency of graph exploration has already been proven [7], a final user study will be conducted to show that the combination of CET and an integrated view on the Personal Information Space is an improvement for Personal Information Management. As an outlook to further research on the topic and for guidelines to first specialised implementations, a set of interviews with different types of users will be conducted to reveal Personal Information Management scenarios and pressing topics when it comes to user-automated tasks that should better be fulfilled by software.

VIII. Progress of the Thesis

In this section the current progress of the thesis is described for each aspect.

The *integration* concept has been presented in [11] and is currently integrated in the CET as a successful proof-of-concept. However, the mediation part will be externalised to be independent. Support for disambiguation and information updates are pending development. The communication libraries GLUE and MOCCA have been published in [8] and are used in several projects within the DKE Group. The current development version is 0.3 (which is the third revision) and is expected to be finished soon. Requirements and concepts for persistent resource identification have been discussed in [9] and are ready for files and e-mails. The identification of contacts is still in the concept phase.

The *Creative Exploration Toolkit* has been developed during the BISON project[21]. Since then it has been used for different data sources during demonstrations, for example the CeBIT fair. The graph interaction methods described in this section have been demonstrated during the *ECML PKDD 2010* demo session [10] and are published in [7]. However, complex nodes are still an open issue.

Advanced operations, other than those provided in the CET will be implemented for validation purposes when the integrated view is finished.

The *validation* is in an early phase. The benefits of structural views in exploratory search have been proven with a user study on the CET, published in [7], Chapter 5. It has been clearly shown that exploration tasks regarding relationships between information elements or emerging structures could be solved much faster with the CET than by the control group using a Web Browser. As a conclusion the graph view is a relevant improvement. However, validation of the remaining thesis aspects is still pending.

IX. Conclusion

Providing an integrated view on Personal Information without excluding existing infrastructure or additional devices has proven to be a difficult task. While the vision still points towards software support on a semantic level, enabling technologies must be developed. This thesis concentrates on the questions of transparent integration and exploratory graph search in the Personal Information Space, where information is highly volatile and will be accessed not only from different applications, but also from different devices simultaneously.

Certain aspects had to be excluded from the thesis at all:
Security and privacy are a highly relevant topic nowadays. However, implementing a complete set of security mechanisms for the envisioned system, let alone in an open context such as the Internet, would be a thesis on its own. As a result, the prototype will be restricted to single-user access in a secure environment.
The same holds for collaboration. Not only does a multi-user context imply a hardened security implementation, it also opens up a whole set of additional research questions which cannot be regarded in the course of this thesis.

Eventually, the results of this thesis are meant to improve everyday interaction with personal information and enable new technologies for software support in user-automated interaction to free the user's mind for the real task at hand and enable novel applications.

[20] https://www.cs.cmu.edu/~enron/

[21] BISON – Bisociation Networks for Creative Information Discovery, FP7-ICT-2007-C FET-Open, con- tract no. BISON-211898 (http://www.bisonet.eu)

References

[1] W. Akhtar, J. Kopechky, T. Krennwallner, and A. Polleres, "XPARQL: Traveling between the XML and RDF worlds – and Avoiding the XSLT Pilgrimage," in *The Semantic Web: Research and Applications*, vol. Volume 5021/2008, Springer Berlin / Heidelberg, 2008, pp. 432–447.

[2] T. Berners-Lee, R Fielding, and L Masinter, "Uniform Resource Identifier (URI): Generic Syntax," RFC 3986, Jan. 2005. [Online]. Available: http://tools.ietf.org/html/rfc3986.

[3] T. Berners-Lee, L. Masinter, and M McCahill, "Uniform Resource Locators (URL)," RFC 1738, Dec. 1994. [Online]. Available: http://tools.ietf.org/html/rfc1738.

[4] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data-the story so far," *International journal on semantic web and information systems*, vol. 5, no. 3, pp. 1–22, 2009.

[5] R. Boardman, "Improving Tool Support for Personal Information Management," PhD thesis, University of London, 2004.

[6] E. R. Gansner and Y. Hu, "Efficient, Proximity-Preserving Node Overlap Removal," *J. Graph Algorithms Appl.*, vol. 14, no. 1, pp. 53–74, 2010.

[7] S. Haun, T. Gossen, A. Nürnberger, T. Kötter, K. Thiel, and M. Berthold, "On the Integration of Graph Exploration and Data Analysis: The Creative Exploration Toolkit," in *Bisociative Knowledge Discovery*, M. R. Berthold, Ed., ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, vol. 7250, pp. 301–312, ISBN: 978-3-642-31829-0. DOI: 10.1007/978-3-642-31830-6_21.

[8] S. Haun, R. Krüger, and P. Wehner, "SENSE: Combining Mashup and HSM technology by semantic means to improve usability and performance," in *Online Communities: Enterprise Networks, Open Education and Global Communication: 16. Workshop GeNeMe '13*, T. Köhler and N. Kahnwald, Eds., Gemeinschaften in Neuen Medien, Dresden: TUDpress, 2013, pp. 61–72.

[Online]. Available: http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-125715.

[9] S. Haun and A. Nürnberger, "Towards Persistent Identification of Resources in Personal Information Management," in *SDA*, L. Predoiu, A. Mitschick, A. Nürnberger, T. Risse, and S. Ross, Eds., ser. CEUR Workshop Proceedings, vol. 1091, CEUR-WS.org, 2013, pp. 73–80. [Online]. Available: http://dblp.uni-trier.de/db/conf/ercimdl/sda2013.html#HaunN13.

[10] S. Haun, A. Nürnberger, T. Kötter, K. Thiel, and M. Berthold, "CET: A Tool for Creative Exploration of Graphs," in *Machine Learning and Knowledge Discovery in Databases*, ser. Lecture Notes in Computer Science, J. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, Eds., vol. 6323, Springer Berlin Heidelberg, 2010, pp. 587–590, ISBN: 978-3-642-15938-1. DOI: 10.1007/978-3-642-15939-8_39. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15939-8_39.

[11] S. Haun, S. Schulze, and A. Nürnberger, "Towards an update-enabled Mediator System using Semantic Web technology," in *Grundlagen von Datenbanken 2010*, ser. CEUR workshop proceedings, 2010.

[12] I. Herman, G. Melancon, and M. S. Marshall, "Graph visualization and navigation in information visualization: a survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, no. 1, pp. 24 –43, 2000. DOI: http://dx.doi.org/10.1109/2945.841119. [Online]. Available: http://dx.doi.org/10.1109/2945.841119.

[13] Y. Koren and A. Çivril, "The Binary Stress Model for Graph Drawing," in *GD 2008*, I. G. Tollis and M. Patrignani, Eds., ser. LNCS 5417, Springer-Verlag Berlin-Heidelberg, 2009, pp. 193–205.

[14] A. Langegger, W. Wöß, and M. Blöchl, "A Semantic Web Middleware for Virtual Data Integration on the Web," in *The Semantic Web: Research and Applications*, ser. Lecture Notes in Computer Science, vol. Volume 5021/2008, Springer Berlin / Heidelberg, 2008, pp. 493–507. DOI: 10.1007/978-3-540-68234-9_37.

[15] G. Marchionini, "Exploratory Search: From Finding to Understanding," *Commun. ACM*, vol. 49, no. 4, pp. 41–46, Apr. 2006, ISSN: 0001-0782. DOI: 10.1145/1121949.1121979. [Online]. Available: http://doi.acm.org/10.1145/1121949.1121979.

[16] C. Newman, "IMAP URL Scheme," RFC 5092, Nov. 2007. [Online]. Available: http://tools.ietf.org/html/rfc5092.

[17] S. J. Payne, "Mental Models in Human-Computer Interaction," in *The Human-Computer Interaction Handbook*, A. Sears and J. A. Jacko, Eds., Lawrence Erlbaum Associates, 2008, pp. 63–75.

[18] P Resnick, "Internet Message Format," RFC 5322, Oct. 2008. [Online]. Available: http://tools.ietf.org/html/rfc5322.

[19] K.-U. Sattler, I. Geist, and E. Schallehn, "Concept-based querying in mediator systems," *The VLDB Journal*, vol. 14, no. 1, pp. 97–111, 2005. [Online]. Available: http://dblp.uni-trier.de/db/journals/vldb/vldb14.html#SattlerGS05.

[20] D. Spinellis, "Index-Based Persistent Document Identifiers," *Inf. Retr.*, vol. 8, no. 1, pp. 5–24, Jan. 2005, ISSN: 1386-4564. DOI: 10.1023/B:INRT.0000048494.05013.6a.

[21] M. Vander Sande, R. Verborgh, S. Coppens, T. De Nies, P. Debevere, L. De Vocht, P. De Potter, D. Van Deursen, E. Mannens, and R. Van de Walle, "Everything is connected: using linked data for multimedia narration of connections between concepts," eng, in *11th International Semantic Web Conference, Proceedings*, Boston, MA, USA, 2012, p. 4. [Online]. Available: http://iswc2012.semanticweb.org/sites/default/files/paper_10.pdf.

All hyperlinks mentioned in the course of this paper have been verified and were available by the day of submission.

Digitized Forensics: Segmentation of Fingerprint Traces on Non-Planar Surfaces Using 3D CLSM

Stefan Kirst [a, b]
[a] Otto-von-Guericke-University Magdeburg
Universittsplatz 2, 39106 Magdeburg
Email: stefan.kirst@iti.cs.uni-magdeburg.de
[b] University of Applied Sciences in Brandenburg
Magdeburger Str. 50, 14770 Brandenburg an der Havel
Email: kirsts@fh-brandenburg.de

***Abstract*—In digitized forensics the support of investigators in any manner is one of the main goals. Using conservative lifting methods, the detection of traces is done by the investigator himself. Whenever contactless methods are utilized, there is often no preselected area of interest, especially in a coarse scan scenario. Finding traces on challenging surfaces is still quite difficult when it comes to fingerprints. Our approach for detection of fingerprint traces including segmentation aims for the determination of distinctive differences between the surface and the trace. Therefore, we work with an approach based on Clausing et al. using statistical features on gray-level-co-occurrence matrices, surface roughness features and color features in a blockwise manner to segment fingerprint residue and surface. By applying our approach on discriminative surfaces with different latent fingerprints in three different angles we evaluated our approach with a total amount of 28855 feature vectors. We are able to build general models for segmentation of latent fingerprint trace on different substrates using 3D confocal laser scanning microscopy (CLSM) in three different acquisition angles with correctly classify instances up to 96%, allowing a fairly reliable detection by segmentation. Furthermore, the results show differences when comparing the classification results of different angles. Even though, these results show a positive tendency regarding the usability of the proposed methods, further improvements for trace detection on challenging surfaces need to be done in the future.**

I. Introduction

Fingerprints are very important sources of information in the field of criminalistics. However common lifting methods like adhesive foil, sticky tape and powder may alter the trace. Using contactless non-invasive repeatable methods like 3D contactless confocal laser scanning microscopy (CLSM) is an upcoming, integrity preserving solution.

The ascertainability of fingerprint residue not only changes when acquiring different substrates, but also varies when surfaces are non-perpendicularly positioned under the sensor. This is a result of different behavior in contrast or the affinity to outliers in scan data. In addition, there are also challenges that arise from the perspective distortion that occur when acquiring non-planar surfaces. By using a block based approach for the determination and equalization of such distortions we were already able to show significant improvements regarding the relative positioning of fingerprint features [1]. We also addressed the potential of parallelization in intra- and inter-scan scenarios, the integration of provenance information for supporting the chain of custody and the evaluation of the overhead for such integration [2], [3].

Besides the determination and equalization of perspective distortion there remain the challenges of segmentation and detection of finger traces, before one can actually analyze the features (minutiae) of a fingerprint trace. Generally the detection of a trace is done by the investigator, when lifted manually. Especially for large or distributed crime scenes the digital process of analyzing traces gains importance, which makes the detection of traces compulsive. Whenever a trace is acquired from a surface with complex texture, it is hard to distinguish between the trace and the background [4]. This becomes even more challenging when non-planar acquisition angles are used due to shape of the surface or sensor positioning.

The approach presented in this paper deals with the segmentation of latent fingerprints on challenging surfaces without any physical enhancement like powdering. The methodology of detection by segmentation has already been shown effective in [5]. In order to do that, we utilize a subset of features used by Clausing et. al. [6] for the segmentation of toolmarks on lock cylinders. Thereby our classification approach uses statistical texture features, roughness features and naive color data analysis. Challenging surfaces like brushed metal have already been identified in [7] whereupon we created a test set of different challenging surfaces, see Table I. We not only try to segment traces within a single surface, we also want to build general fingerprint models over a variety of substrates and angles. Hence, the main goal of this work is to pinpoint the possibility of building a general model for the segmentation of fingerprints on different substrates using 3D confocal laser microscopy for a detection purpose. For each surface we acquire a set of subscans with a latent fingerprint[1] in three different angles (0°, 10°, 20°) using contactless 3D confocal microscopy. As we extract our features in a blockwise manner we are able to use 28855 feature vectors in the evaluation.

The presented paper is structured as follows: In section II, we summarize relevant related work before we explain our concept for general fingerprint trace detection in section III. In this section, we describe necessary preprocessing steps and the used features. The evaluation of our approach is shown

[1] fingerprints are different on each substrate or, if originated from the same finger, freshly applied on the surface

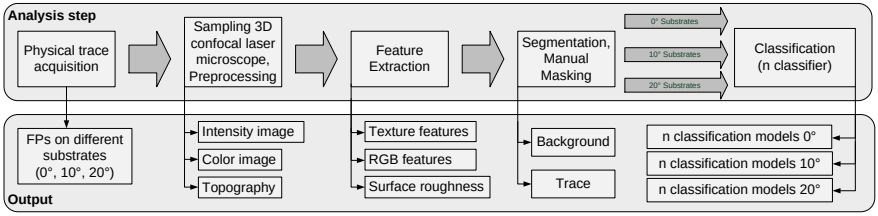


Fig. 1: process model

by presenting our experimental test set and implementation in section IV and the achieved results in section V. In the last step, we summarize our findings in section VI by drawing a conclusion and pointing out future work.

II. Related Work

Makrushin et al. [7] proposes a method to detect latent fingerprints in coarse scans with a comparable set of features but fails in classification for brushed metal and non-glossy car body finish. Since we work on a high resolution level and use color and topography data as well, we expect higher classification results with the utilized feature set. However the coarse scan methodology ensures a trace scan without acquiring any personal data.

The method presented in [6] segments toolmark traces on pins of locking cylinders. Blockwise calculated gray-level-co-occurrence matrices (GLCM) in eight different directions and distances are generated for each scan. Such GLCMs are used to describe the relation of gray values in their neighborhood. As stated in [6], the approach uses a selection of five GLCM-based features (Contrast, Entropy, Correlation, Energy, Homogeneity) as well as surface roughness and color features on a wide selection of classifiers. The results using all 351 features for these methods are up to 98% true positives. We use a subset of these features for segmentation purposes for fingerprints traces.

Also working on topography data, [8] uses a chromatic white light sensor's topography to classify different surfaces. Therefore, it uses profile, waviness and roughness features taken from the international standards issued by the International Organization for Standardization (ISO 4287, ASME B46.1m, ISO 25178-6). Besides the creation of five material classes, it is also pointed out, that some of the features are sensitive to the presence of a latent fingerprint, when comparing the features of surfaces with and without a fingerprint. We chose a selection of eight roughness features from the selection of surface texture features in [8] to adapt them within the present scenario on intensity and topography data, like [6].

According to Oermann et al [9] five different fusion levels exist, whereupon the present approach works as a feature level fusion, since the combination of different raw data (color, intensity, topography) takes place when the feature set is created as presented in subsection III-B. In [10] the topography data is not used on purpose for the detection and localization of fingerprints in low-resolution scans, since the accuracy of the classification process does not improve when using such data. In this paper we use the topography data as we have access to high-resolution scans from our acquisition process. In [11] a match level fusion approach is used to combine topography and intensity data for biometric user authentication as multimodal signal processing, by opposing such data in the validation process. However we concentrate on the blockbased classification process for the segmentation of fingerprint residue on different substrates utilizing a subset of the proposed feature set of [6].

III. Concept

The detection of fingerprint residue as a forensic trace using computational methods is intended to be a support for the investigator. The detection is a crucial step in the analysis of a trace. By using high-resolutional data, of course this approach improves the detection rates of coarse scan scenarios, such as [12], by implementing a detection by segmentation as already proven effective in [6].

Our approach for the detection by segmentation of fingerprints consists of five main steps: physical- and digital acquisition, feature extraction, masking and classification (see Figure 1). In the acquisition we first provide a set of different substrates with different fingerprints, that are then digitized in different angles using a 3D confocal laser microscope. The fingerprints differ in varying angles when the same substrate is used. As a result of the acquisition process we get three perfectly aligned data streams: intensity-, topography- and color data. We provide a test set of different acquisition angles in order to build angle-based models for the segmentation of latent fingerprints on varying substrates. As we use planar and non-planar scans, we are able to exemplarily compare the ascertainability of substrates using a 3D CLSM device.

To successfully segment fingerprints for detection purpose, the raw sensor data needs to be slightly preprocessed to assure meaningful features. After the acquisition and the necessary preprocessing we extract RGB features from the color data, statistical texture features from the intensity data and surface roughness data from intensity and topography data in the feature extraction step. The feature set we are using in our approach is a subset of the feature set proposed by Clausing

et al. [6].
For the creation of a proper ground truth for the classification process a manual selection of trace and surface areas is done according to our best knowledge. Furthermore we apply a selection of different classifier classes as proposed in [5] to avoid a possible overfitting as well as most suitable classifiers for the present challenge. The feature vectors that are created by the feature extraction process for each surface and angle are then combined into angle based models ($0°$, $10°$ and $20°$) using a set of classifiers. To show the potential for an overall model the feature vectors of all scans are combined and classified as well.

By analyzing different angle-based models we not only like to show the feasibility of the approach but to indicate the limits of ascertainability of traces under different environmental factors, such as kind of substrate and angle of acquisition. Therefore it is important to mention the differences in contrast between trace and substrate depending on substrate and angle, which have a big influence on the model building. With functional segmentation models, a detection by segmentation like used in [5] can be done. However, we are aware of the privacy issues, that arise from detecting fingerprints in a non-coarse-scan scenario [12]. Nevertheless, the determination of fingerprint areas in high-resolutional data is of interest, whenever a detection in a coarse-scan scenario is inaccurate or fails.

A. Preprocessing

Except for the topography data, where we apply a plane subtraction, we do not pre-process our data. Moreover, a removal of outliers is not intended, because they only occur in cases of highly reflective, non-planar or very diffuse, light absorbing texture. Both possibilities indicate a characteristic of the surface or the trace and are therefore valuable. Unlike [5] there are also no curvature induced gradients, which need to be eliminated. The plane subtraction we are applying on the captured topography data is necessary to ensure the information value of absolute features like "highest peak" or "total height" and of course to align the z-ranges of planar- and non-planar-scans. We hereby follow the preprocessing suggestion of [8]. The calculation of an ideal plane is done using the least square problem[2]. The necessity for doing this not only for the non-planar ($10°$, $20°$), but for the planar scans as well, comes from an unpreventable non-ideal specimen positioning under the sensor device, which is compensated by subtracting the calculated ideal plane.

B. Feature Set

For classification purposes we calculate features on color-(24bit RGB), intensity-(16bit) and topography(16bit) data using a feature set of 185 features, which is a subset of the introduced feature set for toolmark detection and segmentation used by Clausing et al. [6].

To analyze the intensity data, we use five statistical texture features on gray-level-co-occurrence matrices. The used statistical texture features are *Contrast, Entropy, Correlation, Energy* and *Homogeneity*. For further information on how we calculate these features see [5]. A GLCM is a matrix representation of the distribution of present gray values in the neighborhood within a given offset. The calculation is done for every block of the intensity data in eight directions and four different distances. The statistical features are afterwards calculated on the resulting 32 GLCMs for each block, resulting in 160 statistical texture features for the intensity data. In contrast to [6], these features are only calculated for the intensity data.

The intensity and topography data is furthermore analyzed by calculating roughness features, like suggested in [8]. They are not calculated on GLCMs like the statistical textures features for the intensity data, but still applied on every block. We used a selection of eight roughness features, which are calculated according to the ISO 4287/2000 standard:

1) Arithmetical mean deviation
2) Root mean square deviation
3) Kurtosis of the assessed profile
4) Skewness of the assessed profile
5) Lowest valley
6) Highest peak
7) The total height of the profile
8) Mean Height of Profile Irregularities

The eight surface roughness features are applied on both, intensity and topography data, which adds another 16 features to the feature vector.

For the color data there are three naive features, *minimum, maximum* and *average color*, for each RGB-channel. Those last nine features finalize the feature set and add it up to 185.

The mentioned feature set above is experimental for the intended purpose and is evaluated in the next section.

IV. Test Set and Implementation

In order to evaluate our presented method we use a classification approach of a typical two-class problem (trace vs. no trace). As we try to find a general and substrate independent model for fingerprint traces using a confocal laser microscope, we select a variety of surfaces for our test set. The determination of the mentioned feature vectors (see subsection III-B) are done blockwise and initially for every acquired substrate separately. Afterwards the resulting instances are merged to six different models including a prior application of a Principal Component Analysis for each acquisition angle: $model_{0°}$, $model_{10\_°}$, $model_{20°}$, $model_{0\_PCA}$, $model_{10\_PCA}$ and $model_{20\_PCA}$. By analyzing the true positive and true negative rates as correctly classified blocks in addition with kappa statistics to measure inter-rater agreement [13], we want to show the practicability of our approach.

A. Building the Test Set

To evaluate our approach we decided to use the following surfaces of challenging and cooperative substrates. Each surfaces is acquired using a number of subscans[3]. We also used different fingerprints and fingers on each substrate. The enumeration of fingers starts with the little finger (finger 1) of

[2] using the ImageJ plugin "'Nonuniform Background Removal"' by Cory Quammen, cquammen@cs.unc.edu

[3] subscans are stitched using the Keyence Image Assembler [14]

surface	0° used finger/ #scans	10° fingerprint/ #scans	20° fingerprint/ #scans
hard disc drive platter	finger 2/ 25	finger 3/ 12	finger 8/ 12
matte aluminum foil	finger 10/ 12	-	finger 2/ 12
oak furniture veneer	finger 7/ 25	-	-
white furniture veneer	finger 8/ 16	finger 9/ 12	-
green car paint	finger 7/ 20	finger 7/ 12	finger 10/ 12
glossy aluminum foil	finger 8/ 25	-	-
brushed metal	finger 7/ 25	-	-
glossy black plastic	-	finger 6/ 12	-

TABLE I: test set used for evaluation, collectively consisting of a 232 scans

the left hand and ends with the little finger of the right hand (finger 10). The resulting test set is shown in Table I.

The selection of surfaces[4] mainly addresses the challenge of segmenting latent fingerprints, with varying complexity for different substrates. Furthermore, this selection is not intended to be a comprehensive representation of all crime scene relevant surfaces, but rather an exemplary collection of different surface complexities. Thereby, the hard disc drive (HDD) platter and the matte aluminum foil seem to be the most cooperative surfaces, whereas brushed metal and glossy aluminum foil emerged as most challenging surfaces for planar scans. For the non-planar scans several substrates result in scans with barely visible finger traces, which are are either not with reasonable certainty discriminable from the surface or not discriminable at all[5].

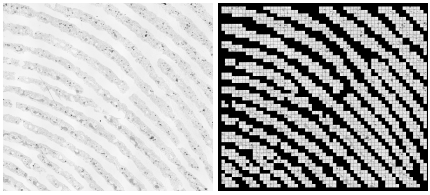
As we learned from [16] latent fingerprints change the most during the first 24 hours due to aging effects. As the acquisition process for a single fingerprint may exceed that time depending on the acquisition angle, all fingerprints were placed on the substrates at least 24 hours before the scan. This decreases the influence of the alteration within the acquisition of a single trace. In matters of scan areas for acquiring fingerprint traces, the ratio of fingerprint residue and surface is intend to be acquired fairly equally.

Since residues of fingerprints are not always from the same thickness, there is more likely a greater amount of ridge areas then valleys. Therefore, the distribution of instances is varying due to differences in pressure and finger movement when placing such traces, even if they come from the very same finger. This results in changes of ridge thickness, which affects the number of trace marked instances. The average distribution is about 3 to 2. The number of instances for every substrate is distributed between 1700 and 2500.

It is to be mentioned that the creation of a non-planar test set is extensive. Acquisition time may easily exceed 10 hours, especially for greater acquisition angles.

B. Ground Truth

The masking of the present data is done manually in a prior step leading to the two-class-problem: trace vs. no trace. As we like to use a single process to mask the present data, we cannot use automated approaches due to very challenging substrate such as brushed metal. We understand that this manual component introduces a point of failure, but we ensure that this was done to our best knowledge and belief. The labeling of all blocks for every scan is done in "marked as trace" and "marked as no trace". Hereby a block is marked whenever at least half of it is covered with the respective trace, which so far seems to work best within the presented approach. In this case segmentation of a fingerprint means the differentiation between ridge lines and the underlying surface. Examples of such block masks are presented in Figure 2. Each feature vector is based on the resulting mask and the in subsection III-B introduced blockwise extracted features.



(a) intensity raw data (b) block mask of Figure 2a

Fig. 2: comparison of raw image and segmented fingerprint mask of a partial latent fingerprint on a hard disc drive platter captured with Keyence CLSM VK-X 110 [14]

For digital data acquisition purpose we used the Keyence VK-X 110 or VK-X 105 confocal 3D laser scanning microscopes (CLSM) [14]. Both microscopes only differ in their mounting device. Sensor parameters are the same for each scan: resolution (1024x768), z-pitch (1µm) and magnification (10x). Only the laser gain and the z-boundaries have to be applied for each scan due to differences between substrate reflectance, trace positioning and acquisition angle.

The used block size is chosen respectively to the ridge thickness and the resolution[6] of the test set. Due to different pressure and finger movement when applying traces on the substrates as well the usage of different fingers, result in different ridge and valley thickness. As we like to use a single block size only, we adapt the generally known Nyquist-Shannon sampling theorem, saying that at a block represents at most half of the smallest trace feature. As stated before the trace feature (in this segmentation case: valleys and ridges)

[4] derived from [7], [15]

[5] not applicable on 10°: aluminum foil(no visibility), oak furniture veneer(barely visible); not applicable on 20°: aluminum foil(no visibility), oak furniture veneer(no visibility)

[6] all concatenations of subscans for each surface and angle have a resolution of about 9694ppi

	$\mathbf{model_0}$	$\mathbf{model_{0\_PCA}}$	$\mathbf{model_{10}}$	$\mathbf{model_{10\_PCA}}$	$\mathbf{model_{20}}$	$\mathbf{model_{20\_PCA}}$
1st best classifier	**90,66% / kappa 0,788** (RotationForrest)	**91,13% / kappa 0,796** (RandomCommittee)	**90,61% / kappa 0,810** (Bagging)	**96,48% / kappa 0,929** (RotationForrest)	**84,96%/ kappa 0,699** (RotationForrest)	**90,89%/ kappa 0,817** (RotationForrest)
2nd best classifier	**90,65% / kappa 0,788** (Bagging)	**90,58% / kappa 0,784** (Bagging)	**90,54% / kappa 0,809** (RotationForrest)	**96,09% / kappa 0,921** (RandomCommittee)	**84,51%/ kappa 0,690** (Bagging)	**89,97%/ kappa 0,799** (RandomForest)
3rd best classifier	**90,35% / kappa 0,780** (RandomSubSpace)	**90,50% / kappa 0,780** (RandomForest)	**90,51% / kappa 0,808** (RandomSubSpace)	**96,06% / kappa 0,920** (RandomForest)	**84,41%/ kappa 0,688** (RandomSubSpace)	**89,76%/ kappa 0,795** (RandomCommittee)

TABLE II: the three best classification results for each acquisition angle

may differ. The test set presented in Table I shows that a block size of 32x32 fulfills those requirements and works with the presented feature set (see subsection III-B) at the same time.

C. Classification

In total an amount of 28855 distinct feature vectors is used for testing and evaluation.

For classification of the present two-class problem we used the WEKA[7] [13]. [17] states, that the choice of a classifier is done depending on the specific application it is used for, since an all-purpose classifier does not exist. However, to avoid a possible overfitting for the used classifiers we use a selection of classifiers from different classes. We are using the very same set of classifiers regarding to [5]:

- Bayes: Naive bayes, BayesNet
- functional: Simple Logistic, SMO, RBF Network
- lazy: IB1, KStar
- meta: Bagging, Random Committee, Random Subspace, Rotation Forest
- rule-based: Decision Table, OneR
- tree-based: J48, Random Forest, Random Tree

All classifiers are used with default settings and a 10 fold cross-validation. Analog to [5] we are also using a Principal Component Analysis [18], due to a probable strong feature correlation, to enhance the classification results. The resulting feature set has an average total of 11 features of uncorrelated linear combinations to harden the following classification step (range: from 9 to 13).

Since one of our detection goals is to directly analyze the segmented trace, both True Positive (TP) and True Negative (TN) rates matter. A segmented trace with lots of surface area left in the area of interest is as insufficient for further analysis as a not entirely segmented trace. We combine both rates as a total correctly classified rate, like presented in Table II. We also provide the kappa-statistics for each classification to give a hint on the reliability of the achieved results.

V. Result of our Approach

The results for the fingerprint models for 0°, 10° and 20° and their PCA-versions are presented in Table II with their three best classifiers and the respective kappa values. The rating of the classifiers are based on correctly classified instances. All values are rounded down.

[7]Waikato Environment for Knowledge Analysis - machine learning software, version 3.6.8, online available: http://www.cs.waikato.ac.nz/ml/weka/

The outcome shows a rate of correctly classified instances from at least 84% (PCA: at least 89%). The $model_{10}$ indicates a slightly better acertainability when looking at the classification results (90%, PCA: 96%). We are aware that the comparison is not clear, as all models results are respectively to their unique underlying test set, see Table I. The $model_{20}$ shows the worst results, even if the PCA results are fairly the same, when compared to the planar model. This probably comes from the more limited test set and the overall worse sensor performance at greater acquisition.

However the results are promising, since we also build an overall model over all angles ans substrates with classification results of 89% with kappa 0,78 using Bagging (PCA: 92%, kappa 0,84, RotationForest). Although, the ratio of instances between the acquisition angles are not evenly distributed, the confirmation of classification results compared to the planar model pinpoints the feasibility of the presented approach. Furthermore we calculated a model over all acquisition angles based on substrate all models share and achieved positive classification results of 91% with kappa 0,83 using RotationForest (PCA: 95%, kappa 0,90, RotationForest).

As stated, comparing those models is not appropriate, as they do not share the exact same basis of substrates, see Table I. Nevertheless, for each angle and their test sets it was possible to find a general classification model for the detection by segmentation of latent fingerprints using 3D confocal laser microscopy. Furthermore, all meta classifier show the best results overall with the present data, regardless of an applied PCA.

As it is hard to compare the angle based models, one can not judge over the overall model either, since the underlying angle based models have not the same ratio within the final model. Nevertheless it is a promising glimpse to an actual generalized model for the segmentation of fingerprints using 3D confocal laser microscopy.

VI. Conclusion and Future Work

As shown above, the detection by segmentation of fingerprint traces using general angle-based models for different substrates on data acquired with 3D confocal laser microscopy is a promising endeavor. By using statistical texture features on blockwise calculated gray-level-co-occurrence matrices of the intensity data, the roughness analysis on intensity and topography data as well as the analysis of the color data for each RGB-channel, we are able to provide a fairly reliable and promising segmentation method. The presented approach shows classification rates up to 91% with kappa values up to 0,79 when building general angle-based classification models. Even if a comparison of angle-based models is inappropiate due to differing sizes of test sets, the small amount of reduction

of the classification rates is mentionable, because increasing acquisition angles are accompanied with increasing noise and a greater amount of outliers.

When using a Principal Component Analysis, the results of up to 96% (kappa 0,92) especially for non-planar models propose the analysis of the possibility of non-planar acquisiton angles as an optimal setup when scanning traces. Also, the influence of distortion due to a non-perpendicular sensor perspective needs to be investigated. As we work within a fine-scan scenario, there remain privacy issues, that have to be dealt with in order to preserve personal rights [12]. A solution may be an analysis of the application of the feature set on coarse scan scenarios. By building more homogeneous models over all acquisition angles, more meaningful angle-independent models may arise. Furthermore, performing a proper feature selection and expanding the feature set aswell as the test set should increased the results. A minor incease of the results by altering the classifier‘ parameters is plausible as well. Finally, the evaluation of this approach as an actual segmentation process for biometric purposes using very high-resolutional data and tools like from NBIS[8] for minutiae extraction and matching is pending.

ACKNOWLEDGMENT

The author would like to thank Jana Dittmann and Claus Vielhauer for valuable comments, supervising this research and for overall supporting our work. Furthermore the author thanks Christian Arndt and Eric Clausing for numerous fruitful discussions. The work in this paper has been funded in part by the German Federal Ministry of Education and Science (BMBF) through the Research Program under Contract No. FKZ:13N10818 and FKZ:13N10816.

REFERENCES

[1] S. Kirst, E. Clausing, J. Dittmann, and C. Vielhauer, “A first approach to the detection and equalization of distorted latent fingerprints and microtraces on non-planar surfaces with confocal laser microscopy,” *Proc. SPIE*, vol. 8546, pp. 85 460A–85 460A–12, 2012.

[2] S. Kirst and M. Schäler, “Database and data management requirements for equalization of contactless acquired traces for forensic purposes,” *Workshop on Databases in Biometrics, Forensics and Security Applications (DBforBFS)*, vol. BTW-Workshops, pp. pages 8998, Køllen–Verlag, 2013.

[3] ——, “Database and data management requirements for equalization of contactless acquired traces for forensic purposes - provenance and performance,” *Datenbank-Spektrum*, vol. 13, no. 3, pp. 201–211, 2013.

[4] A. Makrushin, T. Kiertscher, R. Fischer, S. Gruhn, C. Vielhauer, and J. Dittmann, “Computer-aided contact-less localization of latent fingerprints in low-resolution cwl scans,” in *Communications and Multimedia Security*, ser. Lecture Notes in Computer Science, B. Decker and D. Chadwick, Eds. Springer Berlin Heidelberg, 2012, vol. 7394, pp. 89–98.

[5] E. Clausing, C. Kraetzer, J. Dittmann, and C. Vielhauer, “A first approach for the contactless acquisition and automated detection of toolmarks on pins of locking cylinders using 3d confocal microscopy,” in *Proceedings of the on Multimedia and security*, ser. MM&Sec ’12. New York, NY, USA: ACM, 2012, pp. 47–56.

[6] E. Clausing and C. Vielhauer, “Digitized locksmith forensics: automated detection and segmentation of toolmarks on highly structured surfaces,” pp. 90 280W–90 280W–13, 2014.

[8] “NIST Biometric Image Software” - see [online]: http://www.nist.gov/itl/iad/ig/nbis.cfm

[7] A. Makrushin, M. Hildebrandt, R. Fischer, T. Kiertscher, J. Dittmann, and C. Vielhauer, “Advanced techniques for latent fingerprint detection and validation using a cwl device,” pp. 84 360V–84 360V–12, 2012.

[8] S. Gruhn and C. Vielhauer, “Surface classification and detection of latent fingerprints: Novel approach based on surface texture parameters,” in *Image and Signal Processing and Analysis (ISPA), 2011 7th International Symposium on*, sept. 2011, pp. 678 – 683.

[9] A. Oermann, T. Scheidat, C. Vielhauer, and J. Dittmann, “Semantic fusion for biometric user authentication as multimodal signal processing,” in *Multimedia Content Representation, Classification and Security*, ser. Lecture Notes in Computer Science, B. Gunsel, A. Jain, A. Tekalp, and B. Sankur, Eds. Springer Berlin Heidelberg, 2006, vol. 4105, pp. 546–553.

[10] A. Makrushin, M. Hildebrandt, R. Fischer, T. Kiertscher, J. Dittmann, and C. Vielhauer, “Advanced techniques for latent fingerprint detection and validation using a cwl device,” pp. 84 360V–84 360V–12, 2012.

[11] A. Makrushin, T. Kiertscher, M. Hildebrandt, J. Dittmann, and C. Vielhauer, “Visibility enhancement and validation of segmented latent fingerprints in crime scene forensics,” pp. 866 508–866 508–12, 2013.

[12] M. Hildebrandt, J. Dittmann, M. Pocs, M. Ulrich, R. Merkel, and T. Fries, “Privacy preserving challenges: new design aspects for latent fingerprint detection systems with contact-less sensors for future preventive applications in airport luggage handling,” in *Proceedings of the COST 2101 European conference on Biometrics and ID management*, ser. BioID’11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 286–298.

[13] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining Practical Machine Learning Tools and Techniques*, 3rd ed. Elsevier, 2011.

[14] Keyence Corporation, “Vk-x100/x200 series 3d laser scanning microscope,” Online available: http://www.keyence.com/products/microscope/microscope/vkx100_200/vkx100_200_specifications_1.php, last checked 18/06/2014, 2013.

[15] M. Hildebrandt, R. Merkel, M. Leich, S. Kiltz, J. Dittmann, and C. Vielhauer, “Benchmarking contact-less surface measurement devices for fingerprint acquisition in forensic investigations: Results for a differential scan approach with a chromatic white light sensor,” in *Digital Signal Processing (DSP), 2011 17th International Conference on*, July 2011, pp. 1–6.

[16] R. Merkel, S. Gruhn, J. Dittmann, C. Vielhauer, and A. Braeutigam, “General fusion approaches for the age determination of latent fingerprint traces: results for 2d and 3d binary pixel feature fusion,” pp. 82 900Y–82 900Y–16, 2012.

[17] R. O. Duda, P. E. Hart, and D. G. Stock, *Pattern Classification*, 2nd ed. Wiley-Interscience, 2000.

[18] G. Dunteman, *Principal Components Analysis*, ser. Quantitative Applications in the Social Sciences. SAGE Publications, 1989, no. Nr. 69.

Methods for predicting crash severity prior to vehicle head-on collisions

Andreas Meier
Driver Assistance and Integrated Safety
Group Research, Volkswagen AG
Email: andreas.meier1@volkswagen.de

***Abstract*—Upcoming safety systems like accident-adaptive restraint systems may help to improve vehicle safety. One major challenge for the realization of these systems is that they may require a fast and interpretable function which predicts the severity of an accident prior to collision. Therefore, only with accident parameters estimated by precrash car sensors the severity of the upcoming collision has to be predicted. In this work, we give an overview of data-driven methods to find classification and regression models for this problem automatically. For that, we preprocess crash simulation data and train different models. We also evaluate their performance and discuss the results. Finally, we finish with a conclusion and research questions, which may lead to an application of these models for future, safer vehicles.**

I. Introduction

Vehicle safety is a very challenging research field for automobile manufacturers because safety requirements are rising continuously. Especially governments and customer organizations like *European New Car Assessment Programme* (Euro NCAP) introduce new safety assessments, e.g. for autonomous emergency braking, leading to new and improved safety systems [1]. Beyond these assessments, the demand of customers for safer vehicles but also the pursuit to satisfy the *Vision Zero*, which seeks to avoid seriously injured or killed people on the road, drive developments in vehicle safety [2].

A possible solution for fulfilling some of these requirements may be accident-adaptive safety systems like advanced airbags [3, 4]. These systems adapt their behavior depending on the collision more specifically than current safety systems do. However, some adaptive systems may require the prediction of crash severity prior to collision to allow a timely adaptation. Hence, the prediction of crash severity must be based solely on accident parameters estimated by precrash sensors like cameras, radar, and so on. Furthermore, these parameters may change prior to collision so that the severity must be predicted within a few hundred milliseconds at maximum.

In this work, we give an overview of two methods which predict the severity of an impending collision by processing estimated accident parameters to allow an adaptation of safety systems. The first method categorizes a collision into one of several classes so that a classification problem must be solved. The second method outputs a *universal* severity measure in a continuous domain so that it corresponds to a regression problem. For both methods we want to find fast and accurate prediction models, which should also be interpretable.

The paper is structured as follows. At first, we explain the background and related literature. Then, we describe our general system approach and cover the classification and regression approaches in the next sections. We finish our paper with a short discussion of the results and a conclusion with future research questions.

II. Background

In this section, we describe the necessary background about vehicle safety and related literature in order to improve the comprehensibility of this work.

A. Vehicle Safety

One of the main objectives of vehicle safety is to protect persons from injuries or death in car accidents. The term *crash severity* can be defined in multiple ways, but we focus on the severity describing the effect of the collision on the vehicle structure [5]. Knowing this severity enables us to adapt safety systems which may reduce the risk of injury of the occupants.

In most today's vehicles, an *electronic control unit* (ECU) detects crashes by classifying signals of crash sensors, which register accelerations of the vehicle structure. In figure 1, we show an acceleration signal and its integral called *velocity curve* for a 20 km/h head-on collision. This velocity curve has been normalized to start at point $(0, 0)$ to describe the change in velocity due to the collision. In contrast to the velocity curve, the acceleration signal is very noisy because of high-frequency signal parts caused by elastic-plastic vibrations, instrumentation noises, etc. [6]. A velocity curve is also advantageous because many crash severity measures can be extracted from it for expressing the severity of the collision [7, 8, 9]. For instance, the maximum change in velocity due to the collision Δv_{max} amounts to -5.4 m/s in figure 1.

Estimating crash severity in order to adapt safety systems solely on crash sensor data is challenging. As Seiffert and Gonter explain, restraint systems must be fired within 30 msec at maximum after the collision has started for protecting the occupants [4, p. 116]. If we consider the velocity curve in figure 1, the remaining curve after 0.03 sec is difficult to estimate so that the crash severity may not be determined. Therefore, today's restraint systems are optimized to handle all crash situations well without optimizations for specific crash situations. Nevertheless, accident-adaptive control of safety systems may offer the potential to improve vehicle safety

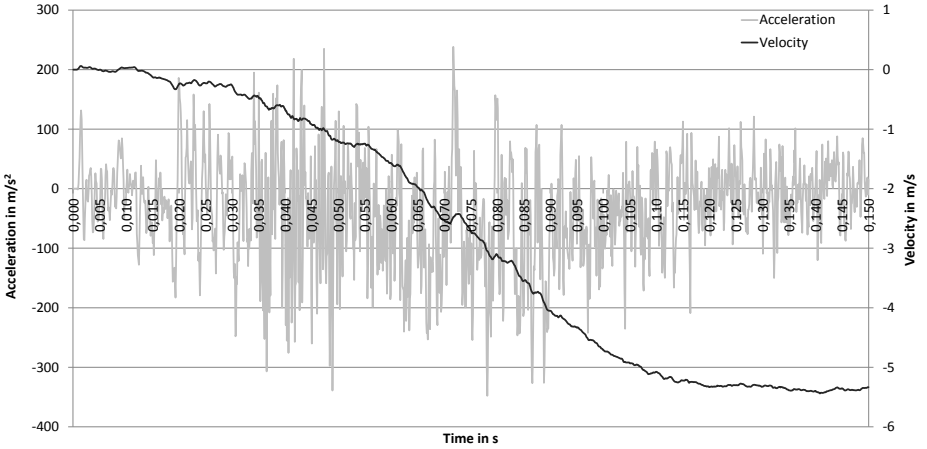


Fig. 1. Acceleration and velocity sensor signals of a 20 km/h head-on collision

further [3]. If the airbag ECU would know the full velocity curve, an improved control of safety systems may be possible. However, these systems often require the prediction of crash severity prior to collision to allow a timely adaptation. Hence, we want to use estimated accident parameters gathered by precrash sensors like cameras, radar, etc. as inputs for the prediction function.

B. Related literature

According to our knowledge, not many previous works exist, which provide ideas for predicting crash severity like we do. Sala and Wang process the signals of two crash sensors with a regression model or an artificial neural network to adapt the dual stage airbag inflation [10]. Due to their usage of crash sensors, the crash severity cannot be predicted prior to collision. Cho et al. use precrash information for improving the robustness of an airbag deployment algorithm [11]. Their algorithm processes radar data and own vehicle data for identifying the crash situation with its probability, the time to crash and a simple crash type discrimination. Bunse et al. describe a system which improves the robustness of airbag deployment algorithms and adapts restraint systems by using precrash information [12]. Since their approach also requires crash sensors, it cannot estimate crash severity prior to collision. Wallner et al. model vehicles as systems of masses, dampers and springs for multibody simulations, which predict a fine-grained crash severity prior to collision [13]. However, this approach cannot handle large angles or low overlaps for colliding vehicles due to the simplified springs and dampers.

III. General System Approach

In this section, we describe the concept of the prediction function, our workflow for finding a crash severity prediction model, the used data and a necessary similarity function.

A. Concept of the prediction function

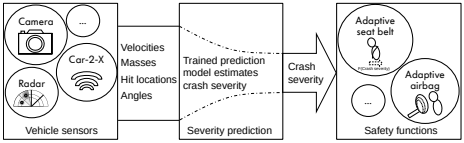


Fig. 2. Concept of the prediction function

In figure 2, we show the concept of the prediction function. Precrash sensors like cameras, radar, etc. estimate the relevant accident parameters prior to an unavoidable collision. Then, the prediction function computes a crash severity prediction from these parameters. This prediction is performed continuously until the collision starts so that changes in the accident parameters are considered. As soon as the collision begins, the prediction function stops and safety systems may adapt their behavior for the predicted crash severity.

As multiple authors describe, crash severity depends on the masses, the velocities and the stiffnesses of the colliding vehicles [14, 15, 16]. In contrast to mass and velocity, the stiffness of a vehicle is more difficult to process because it is very sensitive to the direction of force. We are also not able to estimate the stiffness with precrash sensors. As a consequence,

we use the normalized point of first impact on the vehicle front (*hit location*) and the collision angle as additional parameters.

B. Workflow

The objective is to find a model, which uses accident parameters estimated by precrash sensors to predict the severity of an impending car-to-car crash. The model has to output the severity in less than a few hundred milliseconds with sufficiently high prediction accuracy. The necessary accuracy depends on the safety systems to adapt, but has not been specified yet for the ones in development. Thus, the goal is to find the best possible model. Ideally, the model should also be interpretable so that we can identify all limitations of it.

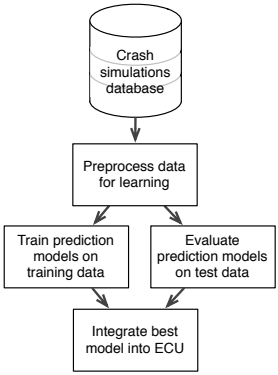


Fig. 3. The system approach to learn a crash severity prediction model

Since these requirements define a challenging problem that is hard to solve by physical modelling, we apply the data-driven approach shown in figure 3. In this workflow, we use crash simulation data, preprocess them and train prediction models with a training data subset. These models are either classification or regression models, which depends on the safety systems to adapt. Whereas restraint systems may need a simple classification, others may require a continuous crash severity measure like Δv. After training, we use a test data subset to evaluate the generalization performance of the prediction models. If we find a model that fulfills all requirements, we may integrate it into an ECU so that it can be used for adapting safety systems in future vehicles.

C. Database

In order to learn a prediction model automatically, we need crash data. Since it is very time-consuming and expensive to gather these data with real crash tests, we use *Finite Element Method* (FEM) simulations. For that, two vehicles, which can be a micro car ($m = 1,184$ kg), a compact car ($m = 1,548$ kg) or an SUV ($m = 2,417$ kg), perform a head-on collision. We also analyzed the database maintained in the *German In-Depth Accident Study* (GIDAS), which investigates serious road accidents in Hanover and Dresden, Germany [17]. Based on this analysis, we were able to identify the relevant domains of our accident parameters. Hence, we vary the velocity of each vehicle between 0 and 64 km/h and the collision angle between 150° and 210°. The hit location is a relative measure depending on the width of each vehicle. -50% denote the outer left corner whereas 50% correspond to the outer right corner so that 0% marks the middle of the vehicle front. We vary the hit location between -80% and 50% in combination with the angle to ensure that only head-on collisions are simulated.

For the FEM simulations, we use the most advanced vehicle crash models available that have been validated for Euro NCAP crash tests. Each simulation covers a time span of 300 msec and we store the velocity curves of each crashed vehicle in our database. We compute the simulations on 64 cores of our high-performance cluster. In spite of these resources, each simulation takes 12 hours to compute so that we were able to perform 173 FEM simulations with 346 crashed vehicles in total. Although more data would be beneficial, even simulation data is difficult to obtain for this problem.

D. Similarity Function

Both learning algorithms require a similarity function for comparing velocity curves. For the classification problem, the simulation data is labeled by comparing their velocity curves with the curves obtained from real wall or barrier crash tests. For the regression problem, the similarity function measures how similar the predicted and the original velocity curves are.

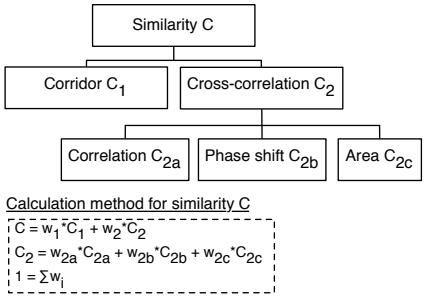


Fig. 4. The ratings of the similarity function based on [9]

In our first experiments, we used Minkowski-based metrics like the Manhattan or Euclidean distance function to compare approximations with their original curves, but these distance functions were not robust enough. The reason is that these distance functions perform pointwise measurements and thus do not consider macroscopic properties like phase shifts or similarity in shape. Therefore, we use our own similarity function shown in figure 4 [9]. This function bases on modified

methods of the software *Correlation and Analysis* (CORA), which is also part of the ISO/TR 16250:2013 standard [18, 19].

This similarity function compares two velocity curves by calculating the weighted average of multiple ratings described by a value between 0 (no similarity) and 1 (perfect match). As the first step, the function sets one curve as reference and limits the upcoming rating calculations to the relevant beginning of the curve. Then, it calculates the corridor rating C_1 by measuring the position of the test curve inside an inner and outer corridor of the reference curve. The closer the test curve is to the inner corridor, the higher the rating so that the corridor rating is actually similar to Minkowski-based metrics. Then, the function calculates the phase shift rating C_{2b} by identifying the best shift between the reference and the test curve for maximizing their cross-correlation. The smaller the shift is, the higher the rating. At optimal shift position, the cross-correlation value describes the correlation rating C_{2a}, which should also be maximal. The area rating C_{2c} calculates the area under each curve and calculates their ratio. The closer this ratio is to 1, the higher the rating. Afterwards the correlation rating C_{2a}, the phase shift rating C_{2b} and the area rating C_{2c} are weighted with their respective factors leading to the cross-correlation rating C_2. Then, this rating and the corridor rating C_1 are also weighted with their respective factors leading to the similarity value C. However, this value C depends on the chosen reference curve so that the symmetry condition of similarity functions is violated. Therefore, we calculate C two times with reference and test curve interchanged. The average of the two C values is our final similarity of the curves.

IV. Crash Severity Prediction as a Classification Problem

In this section, we describe how the crash severity prediction can be solved as a classification problem. For that, we present the preprocessing steps, used algorithms and evaluation results.

In the classification approach, we categorize crash severity into one of several classes. In a previous work, we describe the crash severity of a car in a head-on collision with the most similar barrier or wall crash test of the same car [20]. This is advantageous because safety systems like restraint systems are often evaluated in these crash tests. Thus, predicting the most similar crash test for a car-to-car crash is especially useful for adapting restraint systems because they can be optimized for each crash test explicitly.

A. Data Preprocessing

In our previous work, we describe the labeling of each crashed car of our database with the correct class [20]. We perform this labeling by comparing the velocity curve of a car in a head-on collision with the curves of real barrier or wall crash tests of the same car. For that, we consider up to the first 140 msec of a velocity curve to include all relevant parts of the curve. The crash test with the most similar velocity curve according to the similarity function of section III-D is assigned as class. We label data this way since we cannot compare car-to-car crashes with wall or barrier tests on the basis of accident parameters. Thus, the labeling compares the outputs of different crashes independently of their parameters.

As classes, we use the following crash tests, whose velocity curves are shown in figure 5. Each test is defined by the collision velocity of a vehicle that crashes into the given wall or barrier type with the mentioned overlap of the vehicle front. We choose these crash tests because our still experimental restraint system is optimized for these tests.

- 100% overlap against rigid wall at 27 km/h (FF27)
- 100% overlap against rigid wall at 40 km/h (FF40)
- 100% overlap against rigid wall at 48 km/h (FF48)
- 100% overlap against rigid wall at 56 km/h (FF56)
- 40% overlap against deformable barrier at 40 km/h (ODB40)
- 40% overlap against deformable barrier at 56 km/h (ODB56)
- 40% overlap against deformable barrier at 64 km/h (ODB64)

B. Algorithms

After the data has been labeled, a classification model is trained which maps accident parameters of car-to-car crashes obtained prior to collision to the crash test classes. Each feature vector contains the masses and velocities of both vehicles, the collision angle and the hit locations domains as listed in section III-C. We evaluated the following classification algorithms of the given categories [20]. The algorithms with the highest accuracy of each category are marked in bold.

- Decision trees: BFTree, J48, J48graft, NBTree, SimpleCart, RandomTree, **REPTree**
- Ensemble classifiers: AdaBoost with REPTree, **RandomForest**
- Rule-based systems: **JRip**, Ripple-Down Rule Learner
- Artificial neural networks: MultilayerPerceptron, **RBFClassifier**
- Support vector machine: **C-SVC** (**linear**, polynomial), nu-SVC (linear, polynomial)

Decision trees are classifiers, in which trees evaluate the attributes of an unknown object stepwise with simple logical decisions and predict the object's class. Ensemble classifiers combine multiple weak but simple classifiers to build a more powerful classifier. In rule-based systems, logical rules with attribute evaluations connected by conjunctions or disjunctions reason an unknown object's class. Artificial neural networks take the attributes of an unknown object and use a weighted graph of simple neurons with activation functions to predict the object's class. Support vector machines separate two classes in the data space with a multi-dimensional hyperplane and depending on which side of the hyperplane an unknown object is located, it gets a class. By combining multiple support vector machines more than two classes can be distinguished.

C. Evaluation Results

For evaluating the different classifiers, we use the 173 different FEM crash simulations in our database. 257 of the 346 collided vehicles serve as training set for the classifiers

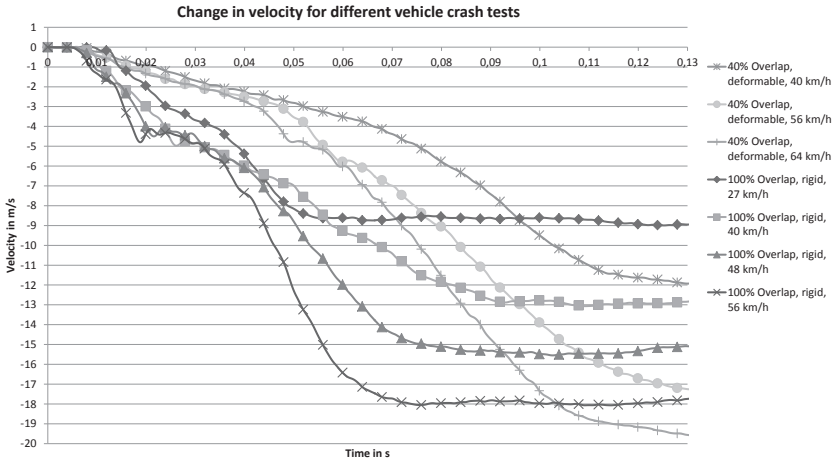


Fig. 5. Velocity curves for different barrier and wall crash tests, source: [20]

whereas the remaining 89 vehicles build the evaluation set. The data sets were created by uniform sampling so that the distribution of instances per class should be similar in both sets. In contrast to the regression approach, we do not create a vehicle-specific but general prediction model. As performance measure, we count the number of correctly classified instances, but we focus on the accuracy on the evaluation set. All algorithms are implemented in *KNIME* and use *WEKA* and *LibSVM* as extensions [21, 22, 23].

TABLE I
CLASSIFICATION ACCURACY ON TRAINING AND EVALUATION SET, SOURCE: [20]

Algorithm	Training set	Evaluation set
REPTree	**256/257 (99.6 %)**	**75/89 (84.3 %)**
RandomForest	**256/257 (99.6 %)**	70/89 (78.7 %)
JRip	224/257 (87.2 %)	64/89 (71.9 %)
RBFClassifier	201/257 (78.2 %)	68/89 (76.4 %)
C-SVC (linear)	218/257 (84.8 %)	67/89 (75.3 %)

In table I, we show the results obtained in our previous work [20]. The REPTree achieves the highest accuracy on the evaluation set and an almost perfect training performance like the RandomForest. The high training performance demonstrates that the training data is likely self-consistent and that the algorithms are able to partition the space correctly. In contrast to the REPTree and the RandomForest, the other listed algorithms achieve a notably lower performance. Considering the prediction time, all algorithms are fast enough because they need 7 msec at maximum for classifying an instance.

V. CRASH SEVERITY PREDICTION AS A REGRESSION PROBLEM

In this section, we describe how the crash severity prediction task is handled as a regression problem. For that, we present the preprocessing steps, used algorithms and evaluation results.

The presented classification approach is especially useful for adapting restraint systems. However, predicting the most similar crash test limits the application of the prediction function because some safety systems may not be able to adapt for this measure. In order to support every possible safety systems, we could either train a specific model for each measure or we could output a *universal* crash severity measure. As already explained, the velocity curve is such a universal measure because various crash severity measures can be extracted from it [9]. Thus, we want to predict the continuous velocity curve resulting in a regression problem. Since this problem is more difficult to solve and more sensitive to the accident parameters than the classification approach, we create a vehicle-specific model. We choose the compact car, because it is the most stable simulation model and our new safety systems are developed for this car. Therefore, the number of usable velocity curves is reduced from 346 to 190.

A. Data Preprocessing

One problem of the regression approach is the large amount of data. Each velocity curve covers a time span of 300 msec with a sampling rate of 10 KHz leading to 3,000 data points per curve. Since the database stores 190 curves, 570,000 data points need to be processed.

In order to simplify the regression problem, we approximate velocity curves because we do not need them at full resolution.

Thus, we evaluated different approximation methods for velocity curves like polynomials of different degrees, Bezier and B-Spline curves [9]. The presented similarity function was used to fit the parameters of each approximation method to each curve. Overall, the B-Spline curve achieves the best fit with an average similarity of 0.976 (standard deviation = 0.008) while still preserving the crash severity measures.

In figure 6, we show a B-Spline approximation for a velocity curve. The shape of the B-Spline curve is defined by four control points of which the first point P_0 is fixed at $(0/0)$. Thus, three control points with two coordinates each remain so that six coordinates in total are necessary. Therefore, we only need a simplified model which predicts these six coordinates instead of 3,000 data points from the accident parameters.

For each curve, we store the B-Spline control points in our database so that the prediction model estimates their coordinates from the corresponding accident parameters.

B. Algorithms

For solving a regression problem, many different algorithms like artificial neural networks, support vector regression and so on exist. However, black-box models are disadvantageous for our problem because we must be able to identify limitations of such a safety-critical function. Thus, regression models expressed as terms are ideal because we can analyze them mathematically. However, for parametric regression like fitting coefficients of a polynomial model, we do not know the underlying function mapping accident parameters to the curve.

In order to find such an interpretable model, we use a technique called *Symbolic Regression*. Symbolic Regression gained significant attention after Koza used it as an example for *Genetic Programming*, which seeks to learn programs automatically [25]. In Symbolic Regression the learning algorithm does not only fit coefficients of a model but learns the model, too. For that, we provide the algorithm our accident parameters as variables and give the basic mathematical operations $+$, $-$, $/$, $*$ but also the functions absolute value, square root and exponential function as operators. Then, the algorithm combines variables and constants with the given operators to create mathematical formulae which minimize a cost function. In that way, the algorithm learns a complete model just from simple building blocks. In our implementation, the model should map the same feature vector as for the classification problem to the B-Spline control point coordinates so that the full approximated velocity curve can be predicted.

C. Evaluation Results

In our previous work, we evaluated the performance of *Cartesian Genetic Programming* (CGP) and *Prioritized Grammar Enumeration* (PGE) [24]. CGP uses a genetic algorithm to recombine and mutate individuals which describe a mathematical formula each. Instead, PGE is fully deterministic and uses Pareto queues and simplification algorithms to perform a brute-force like search in the solution space. For training, we use 143 velocity curves whereas the remaining 47 curves form the test set. We evaluated the performance by measuring the average similarity between the predicted B-Spline curves and the original velocity curves on the training and test set. For training and testing, we always consider the full velocity curve, but the similarity function of section III-D may limit its comparison to a shorter interval than 300 msec.

TABLE II
AVERAGE PERFORMANCE AND STANDARD DEVIATIONS OF LEARNED MODELS, SOURCE: [24]

Algorithm	Training set	Test set
Best CGP	0.607 ± 0.117	0.606 ± 0.118
Best PGE	$\mathbf{0.800 \pm 0.118}$	$\mathbf{0.805 \pm 0.123}$

In table II, we show a comparison between CGP and PGE [24]. Although standard deviations are similar, PGE outperforms CGP notably with an average similarity of 0.8 (80%). In equation 1, we show the six functions found by PGE that calculate the B-Spline control points $P_1(x_1, y_1)$, $P_2(x_2, y_2)$ and $P_3(x_3, y_3)$. As visible, PGE combined the closing velocity $v_{relative}$, the masses m_1 and m_2, the collision angles α_1 and α_2 and the points of impact p_1 and p_2 to mathematical functions. These functions in combination with the B-Spline curve constructing method are sufficient to predict the full velocity curve from estimated accident parameters.

$$
\begin{aligned}
x_1 &= 0.00225 + \frac{0.003684 * |p_2|}{v_{relative}} + 0.0006 * \sqrt{m_2} \\
y_1 &= -2.985431 * exp(0.376616 * \cos\alpha_1 * \\
&\quad |p_2 + 9.034628 * |\sin\alpha_1||) \\
x_2 &= 0.000092 * (646.555477 + \frac{m_2 + 57.285781 * |p_2|}{v_{relative}}) \\
y_2 &= -1.960664 + 0.023319 * |p_1| + 0.012447 * \cos\alpha_2 * \\
&\quad v_{relative} * \sqrt{m_2} \\
x_3 &= 0.000082 * (1102.860176 + \frac{m_1 + 88.096884 * |p_1|}{v_{relative}}) \\
y_3 &= -2.179486 + 0.025837 * |p_1| + 0.012183 * \cos\alpha_1 * \\
&\quad v_{relative} * \sqrt{m_2}
\end{aligned}
\tag{1}
$$

VI. DISCUSSION

In section IV, we evaluate classifiers for mapping accident parameters to real wall or barrier crash tests. The evaluation accuracy of about 84% indicates a good result although it should be improved. Yet, we cannot specify the necessary accuracy because the whole system comprising sensors up to the actual safety system needs to be considered. Nevertheless, classifiers based on decision trees outperform even very sophisticated algorithms like support vector machines. We assume that for this classification task many small clusters in solution space exist. Thus, algorithms like trees that partition the space into many small areas are more successful than others trying to create large complex but few partitions. Nevertheless, trees also need 7 msec at maximum for classifying an

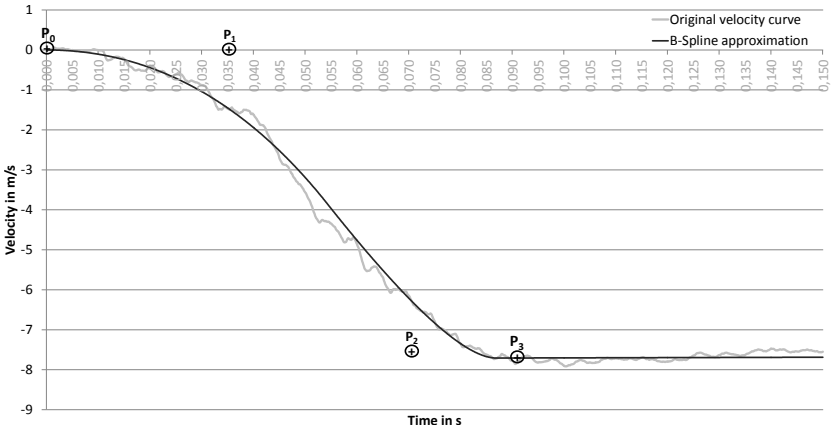


Fig. 6. A velocity curve and its best B-Spline approximation, source: [24]

instance so that they are fast enough for our purpose. Furthermore, they often find interpretable models, but their analysis and evaluating more classification algorithms remain as future research questions. Another question is, whether all chosen seven crash test classes are necessary for adapting the new safety systems. Maybe, the safety systems cannot differentiate between these classes so that the classification problem could be simplified by reducing the number of possible classes.

In section V, we try to solve the crash severity prediction task as regression problem to allow an adaptation of more safety systems. The model found by PGE achieves a good, average prediction performance of 80%. In contrast to the classification approach, defining the desired performance is even more difficult, because more potential safety systems need to be considered. We assume PGE performs better than CGP since PGE may overcome local optima more easily due to its brute-force nature. It is also remarkable that besides approximating velocity curves with B-Splines, we do not provide any expert knowledge but still achieve such a good performance. In general, it would be beneficial to include additional knowledge for improving the performance. However, the large domains of our accident parameters as well as the requirement of estimating the velocity curve make this problem challenging. Hence, we do not have any expert knowledge which improves the performance notably. This is a lack of research which also becomes apparent because we do not know any other comparable solution. Thus, we cannot compare our results to other (non-data-driven) approaches. However, the other advantages of the found model are more obvious. The model satisfies the real-time constraints because it predicts a velocity curve in less than 2 msec on average. It is also interpretable, because we can use the formulae in equation 1 to identify the influence of each parameter on the crash severity analytically. This remains as a future research question. Another question is the acceleration of training since it takes six hours to find a vehicle-specific model with PGE. With faster training, more sophisticated techniques like cross-validation or bootstrapping would become possible.

VII. Conclusion

Improving vehicle safety is getting more and more complex due to increasing safety requirements. One possible solution could be accident-adaptive safety systems, which adapt their behavior on crash severity. However, they often need to know the crash severity prior to collision so that methods are necessary which predict the severity solely on estimated accident parameters. In this work, we give an overview of a classification and a regression approach for solving this problem in a data-driven way. In our experiments, the REPTree classifier scores a good 84% accuracy on our evaluation set. As for the regression approach, we use Symbolic Regression with the PGE algorithm to find mathematical models which predict a universal crash severity with an average performance of 80% on the evaluation set. Besides these good results, the found models in both approaches are likely good to interpret and fast to execute, which are important requirements for such a crash severity prediction function. As future research questions, we plan to improve the performance of both approaches further, analyze the found models and evaluate their benefit in conjunction with a future accident-adaptive safety system. In that way, we hope to enable new and advanced safety systems for future vehicles so that vehicle safety is further improved.

References

[1] Michiel van Ratingen, Aled Williams, Pierre Castaing, Anders Lie, Bernie Frost, Volker Sandner, Raimondo

Sferco, Erwin Segers, and Christoph Weimer. Beyond NCAP: Promoting New Advancements in Safety. In *Proceedings of the 22nd International Technical Conference on the Enhanced Safety of Vehicles*, 2011.

[2] Claes Tingvall and Narelle Haworth. Vision Zero: an Ethical Approach to Safety and Mobility. In *6th ITE International Conference Road Safety & Traffic Enforcement: Beyond 2000*, pages 6–7, 1999.

[3] C. Schramm, F. Fürst, M. van den Hove, and M. Gonter. Adaptive Restraint Systems - The Restraint Systems of the Future. In *Proceedings of 8th International Symposium Airbag 2006*, 2006.

[4] Ulrich W. Seiffert and Mark Gonter. *Integrated Automotive Safety Handbook*. SAE International, 2013.

[5] Peter Felix Niederer, Felix Walz, Markus Hugo Muser, and Ulrich Zollinger. What is a Severe, What is a Minor Traffic Accident? [Original title: Was ist ein "schwerer", was ist ein "leichter" Verkehrsunfall?]. *Schweizerische Ärztezeitung*, 82(28):1535–1539, 2001.

[6] Clifford C. Chou, Jialiang Le, Ping Chen, and Dj Bauch. Development of CAE Simulated Crash Pulses for Airbag Sensor Algorithm/Calibration in Frontal Impacts. In *17th International Technical Conference on the Enhanced safety of Vehicles (ESV), Amsterdam*, 2001.

[7] Lars Kübler, Simon Gargallo, and Konrad Elsäßer. Characterization and Evaluation of Frontal Crash Pulses with Respect to Occupant Safety. In *Airbag. 9th International Symposium and Exhibition on Sophisticated Car Occupant Safety Systems*. ICT, 2008.

[8] Joseph C. Marsh IV, Kenneth L. Campbell, and Upendra Shah. A Review and Investigation of Better Crash Severity Measures: An Annotated Bibliography. Technical report, Highway Safety Research Institute, The University of Michigan, 1977.

[9] Andreas Meier, Mark Gonter, and Rudolf Kruse. Approximation Methods for Velocity Curves Caused by Collisions [Original title: Approximationsverfahren für kollisionsbedingte Geschwindigkeitskurven]. In *Proceedings of 23th Workshop Computational Intelligence*. KIT Scientific Publishing, 2013.

[10] Dorel M. Sala and Jenne-Tai Wang. Continuously Predicting Crash Severity. *Proceedings of 18th International Technical Conference on the Enhanced Safety of Vehicles*, 5 2003.

[11] Kwanghyun Cho, S. B. Choi, and Hyeongcheol Lee. Design of an Airbag Deployment Algorithm Based on Precrash Information. *IEEE Transactions on Vehicular Technology*, 60(4):1438–1452, 5 2011.

[12] Michael Bunse, Marc Theisen, Alfred Kuttenberger, Jorge Sans Sangorrin, Thorsten Sohnke, Jürgen Hötzel, and Peter Knoll. System Architecture and Algorithm for Advanced Passive Safety by Integration of Surround Sensing Information. Technical report, SAE International, 04 2005.

[13] Daniel Wallner, Arno Eichberger, and Wolfgang Hirschberg. A Novel Control Algorithm for Integration of Active and Passive Vehicle Safety Systems in Frontal Collisions. *Journal of Systemics, Cybernetics and Informatics*, 8(5):6–11, 2010.

[14] Ronald L. Woolley and Alan F. Asay. Crash Pulse and DeltaV Comparisons in a Series of Crash Tests with Similar Damage (BEV, EES). Technical report, SAE International, 04 2008.

[15] Alexjandro Angel and Mark Hickman. Analysis of the Factors Affecting the Severity of Two-Vehicle Crashes. *Ingeniería y Desarollo*, 24:176–194, 2008.

[16] T. Sohnke, Jorge Sans Sangorrin, and J. Hötzel. Adaptable Approach of Precrash Functions. In *5th European Congress on ITS*, 2005.

[17] Dietmar Otte, Christian Krettek, Horst Brunner, and Hans Zwipp. Scientific Approach and Methodology of a New In-depth Investigation Study in Germany called GIDAS. In *Proceedings of 18th International Technical Conference on the Enhanced Safety of Vehicles*, 2003.

[18] Christian Gehre, Heinrich Gades, and Philipp Wernicke. Objective Rating of Signals Using Test and Simulation Responses. In *21st International Technical Conference on the Enhanced Safety of Vehicles Conference (ESV)*, 2009.

[19] International Organization for Standardization. Road Vehicles — Objective Rating Metrics for Dynamic Systems. Standard. ISO/TR 16250:2013, 2013.

[20] Andreas Meier, Mark Gonter, and Rudolf Kruse. Precrash Classification of Car Accidents for Improved Occupant Safety Systems. In *Proceedings of 2nd International Conference on System-Integrated Intelligence: Challenges for Product and Production Engineering*, 2014.

[21] Michael R. Berthold, Nicolas Cebron, Fabian Dill, Thomas R. Gabriel, Tobias Kötter, Thorsten Meinl, Peter Ohl, Kilian Thiel, and Bernd Wiswedel. KNIME - The Konstanz Information Miner: Version 2.0 and Beyond. *ACM SIGKDD Explorations Newsletter*, 11(1):26–31, 2009.

[22] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA Data Mining Software: an Update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[23] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[24] Andreas Meier, Mark Gonter, and Rudolf Kruse. Symbolic Regression for Precrash Accident Severity Prediction. In *Hybrid Artificial Intelligent Systems - 9th International Conference, HAIS 2014*, Lecture Notes in Computer Science. Springer, 2014.

[25] John R. Koza. *Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems*. Stanford University, Department of Computer Science, 1990.