Semi-Automated Footwear Print Retrieval Using Hierarchical Features

Tim vor der Brück

School of Information Technology
Lucerne University of Applied Sciences and Arts
Rotkreuz, Switzerland
Email: tim.vorderbrueck@hslu.ch

Thomas Stadelmann

Forensity AG
Technopark
Root-D4, Switzerland
Email: thomas.stadelmann@forensity.com

***Abstract*—Footwear prints are one of the most commonly found pieces of evidence on crime scenes. They can be used both to connect different crimes and to give important clues to the identity of the culprit. In many cases, these footwear prints are distorted or incomplete, which makes fully automated approaches for their identification and comparison unreliable. Hence, we propose a semi-automated approach, where a representation of key features is obtained manually by forensic experts, the comparison with outsole models from a database is done by a computer. To account for potentially poor quality footwear print pictures, we introduce a hierarchical fuzzy search that ranks the outsole models according to their degree of correspondence with the features of the footwear print. Furthermore, we conducted an evaluation that demonstrated the usefulness of the proposed approach.**

Keywords*–*Footwear print; retrieval; tree edit distance.

I. Introduction

Footwear prints can be secured on almost every crime scene [1] and are daily used by law enforcement authorities for the following purposes [2]:

1) Crime scenes where footwear prints belonging to the same shoe have been secured can be linked and deliver valuable information for forensic intelligence and crime analysis.
2) The outsole design of shoes from arrestees can be compared with the footwear prints from the crime scenes and suspects can be linked to open crimes.
3) Crime scene footwear impressions can be associated with a specific outsole model and deliver helpful information for investigations or support purpose 1 or 2.

Police investigators collect for all three purposes images of footwear prints. In case of a newly committed offense, the police investigators manually compare the footwear prints found at the new crime scene with footwear print images originating from earlier offenses, often collected in cardboard files or binders. For this, they are looking for certain striking patterns or characteristics that the footwear prints have in common. In particular, the following two steps have to be performed (see Figure 1). Step one is feature extraction, where the sole patterns are recognized and encoded into an abstract representation. The second step is the search process, where an abstract representation of an image is compared with other abstract representations from a data collection. Up until now, humans are still more effective than computer algorithms in interpreting images with footwear prints from crime scenes since they can better distinguish sole patterns from the noisy background. Therefore, established computerized footwear systems work with images that are encoded by humans yet [3]. This means human forensic footwear examiners assign predefined codes to the identified features on the image. Afterward, images can be searched by the assigned codes, which build an abstract representation of the sole pattern on the image. With this approach, every image interpretable by an expert can be processed. Albeit, assigning the right code is not an easy task because there is an infeasible amount of pattern designs that are constantly being changed but the predefined codes remain fixed. Thus, it is quite possible that two experts encode the same pattern differently. However, an unambiguous representation is typically an important prerequisite for finding corresponding patterns. Using traditional exact search methods, only footwear models are determined that completely matches the given input. Therefore, police departments have limited the number of people coding the images to assure a common standard. This is feasible if the footwear database only belongs to a small department or the administration of multiple departments is centralized.

The benefit of a common data exchange of footwear print information between different police departments has been recognized already in the early nineties [2]. Different initiatives have taken up this issue during the last couple of years [2] as offenders get more mobile and can only effectively be opposed by cooperation. Besides political and organizational issues [4], one important limitation of sharing information is the unavailability of an efficient search system to find relevant information with reasonable effort in big data collections containing images and outsole patterns nationwide or even internationally.

In this paper, we present a search engine with an advantageous division of labor between human and machine. The feature extraction is accomplished by human experts. This way, also very noisy and distorted images, which are currently only interpretable by humans, can be encoded. To ensure a homogenous classification by different users, the number of features is restricted to a rather small standardized set. The retrieval is accomplished by a fault-tolerant search engine, which ranks all outsole models stored in our database according to their degree of correspondence with the input shoe track. See Figure 2 for the architecture of our proposed approach called *Fast*.

 ISBN: 978-1-61208-740-5

The remainder of this paper is organized as follows. In Section II we give an overview of the current state of the art regarding automated footwear print retrieval. In Section III we describe our proposed fuzzy search, whereas its evaluation is contained in Section IV. Finally, we give a conclusion and an outlook to possible further work in Section V.

II. Related Work

From the late seventies on [5], police departments in different countries introduced computer-assisted footwear systems. What they all have in common is that the footwear print images are encoded by humans. This means the users assign predefined codes to the detected geometric forms on the footwear print image. For most of the systems, self-developed coding schemes are employed [3]. The amount of predefined codes varies between the different systems and most systems work with more than 40 different codes subdivided into different groups. To achieve a better selectivity, several additional coding elements can be found among the different systems [6]:

- Different zones on the outsole that can be encoded separately
- Additional properties for some codes to specify the geometric form they represent (e.g., horizontal/vertical)

The existing coding systems work with filters based on simple string matching. As it is possible that different experts encode the same pattern differently, relevant footwear images can easily be missed by the search [7]. To reduce mistakes during the retrieval process some systems work only with a few basic codes [8]. Gross et al. [9] could easily distinguish 99% of the impressing using only nine design elements types in combination with size-relationship.

In the current research, there are several fully automated approaches for footwear print comparison that make use of image processing. Usually, they aim to detect certain geometrical patterns on the images (for instance by conducting a Hough transform) [4] [10] . These patterns are then leveraged to obtain an abstract footwear representation, which allows for accurate similarity estimation. Quite lately, deep learning methods that require no explicit feature engineering become popular in computer vision and also for footwear comparison. Neal Khosla and Vignesh Venkataraman, for instance, [11] construct neural networks for footwear images by means of the VGGNet classifier (VGG stands for Visual Geometry Group) and estimates their similarity by applying the vector norm on activation value differences of neurons belonging to certain layers of the network.

So far, it is still uncertain if there is a fully automated method that provides good recognition rates on real crime scene data. Many published methods has been evaluated on synthetic data or the dataset of used crime scene marks has not been published. Therefore, it is often not possible to repeat experiments or compare methods using benchmarks. [7] assumes that published results are not reliable because of missing gold standard datasets and evaluation standards.

The main drawbacks of fully-automated methods are the runtime, which can amount up to several seconds for a single comparison, and the difficulty for a forensic footwear expert

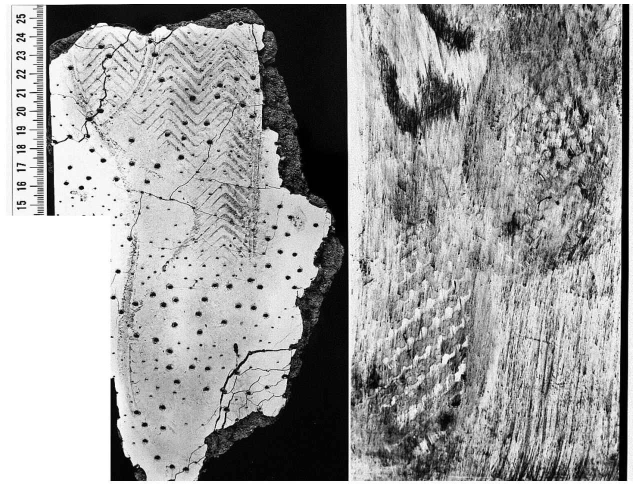

Figure 1. Footwear print from a crime scene.

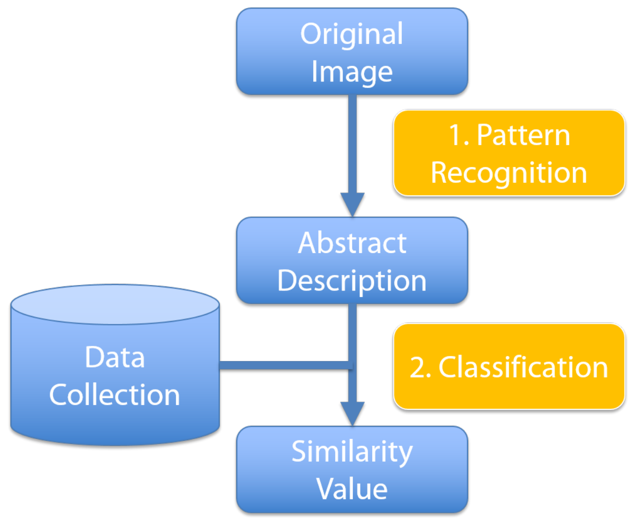

Figure 2. Overview of our proposed methodology for footwear print retrieval.

to take influence in the obtained results. Deep learning-based approaches in contrast are rather fast but behave like black box models and lack interpretability. Furthermore, a neural network has a vast amount of free parameters, which have to be optimized preferably automatically. Such a parameter optimization consumes a large amount of runtime and also needs a lot of training data. This is different with semi-automated approaches like our proposed method, which are however rather rare in academic research. One example in the literature is the method of Gird [2], in which all footwear features values are manually specified and only the search is conducted automatically. Albeit, the proposed method is not error-tolerant and can only establish perfect matches, which considerably hinders its practical usage.

III. Our Approach

In the following, we will first give a rough overview on how our approach works and go into detail afterwards.

A. Overview

For a footwear print found at a crime scene, the following two tasks are of interest:

- Identifying the correct outsole model that is associated with this footwear print
- Comparing two footwear prints from different crime scenes

We will first describe scenario one and cover afterward how to deal with the case that footwear prints should be compared directly with each other.

Basically, a footwear print and its associated outsole model can be represented by a noisy channel model. The informational content of the outsole model is transmitted through a channel that is influenced by environmental conditions and results in a usually distorted and incomplete footwear print. The task of the retrieval system is to obtain the most likely outsole model for the observed print. In practice, this can be a very tedious and difficult task for the following reasons. First, the footwear print could be several hours old already and was potentially affected by weather conditions like rain or wind. Second, only a part of the shoe sole might have had enough contact with the soil to produce a visible mark, which can be particularly the case for hard and dry surfaces. Finally, the sizes of the outsole model and print might be different, which can cause the attribute values not being identically but only proportionally to each other.

Therefore, our approach can establish approximate (fuzzy) matches between shoe tracks and associated outsole models and is error-tolerant in the following ways:

- Different attribute values: Attribute values like *line with* or *circle radius* of a shoe track and its associated outsole model can exhibit minor deviations.
- Different Granularities: The outsole model and the shoe track could be described by different levels of abstraction. Consider for example the case that the outsole model clearly contains an ellipse while the associated shoe track picture is strongly blurred and noisy. The forensic footwear examiner might be unsure whether the picture conveys an ellipse or rather a circle and selects the feature *round shape* instead, which is a hypernym (supertype) of both ellipse and circle.
- Missing features: Not all features contained in the outsole model might be visible at an associated footwear print since the latter potentially conveys only a part of the entire shoe sole. However, if a certain feature shows up in the footwear print, it should also be represented in the outsole model for establishing a match.

B. Tree representation

We represent both the footwear print as well as the outsole models by a tree that describes the observed outsole patterns in a hierarchical way and contains the following type of nodes:

- Feature type: type of a visual shoe sole pattern like lines, round shape, etc. All feature types are ordered hierarchically in a taxonomy tree.
- Feature: an instance of a feature type
- Attribute: property of a feature. Each attribute has a name and an associated value. For example, the attributes for the feature *circle* are *radius* and *line width*. We discern between the following attribute types:
    - Nominal: values of a nominally scaled attribute have no natural ordering and can therefore only be compared for equality.
    - Ordinal scaled: attribute values of an ordinal scaled feature can be enumerated in a natural order. If we compare two different values, one can always decide, which value is smaller and which is larger. However, there might not be a natural origin.
    - Ratio scaled: attribute values of a ratio scale both have a natural ordering and an origin. In addition, the ratio of two attribute values can be interpreted in a natural way, e.g., if this ratio assumes the value of two, then we can conclude that the attribute value belonging to the dividend is twice as high/good/large than the one belonging to the divisor.

An example of such a tree is depicted in Figure 3. A list of all features and associated attributes are given in Table III, the full feature taxonomy is specified in Figure 7. Our employed feature, feature types, and attributes were devised in cooperation with our industry partner, a company specialized in forensics, by a profound study of available outsole models and of existing literature (see for instance [10] for an extensive analysis of outsole patterns) and competing systems.

Note that we provide a browser-based tool to specify the features perceived from the footwear images and which also constructs the associated feature tree automatically. The comparison trees of outsole models are created on the fly for every user query from associated relational database entries. To reduce processing time, we use some kind of prefiltering that rules out entries having low feature agreements with the query tree. In this way, the number of database entries, for which feature trees actually need to be constructed can be considerably reduced and the entire tree creation stage usually takes no more than one second.

C. Obtaining a similarity estimate

To obtain a numerical similarity estimate between a footwear print and an outsole model, we compare their associated trees with each other. There are several tree comparison methods mentioned in the literature, where a selection of them is described below.

Tree Kernel: A tree kernel is a positive-semidefinite similarity measure often employed as SVM (short for support vector machine) kernel (see [12]). Most popular is the common subtree tree kernel, which is based on the assumption that two trees are similar if they contain a lot of common subtrees.

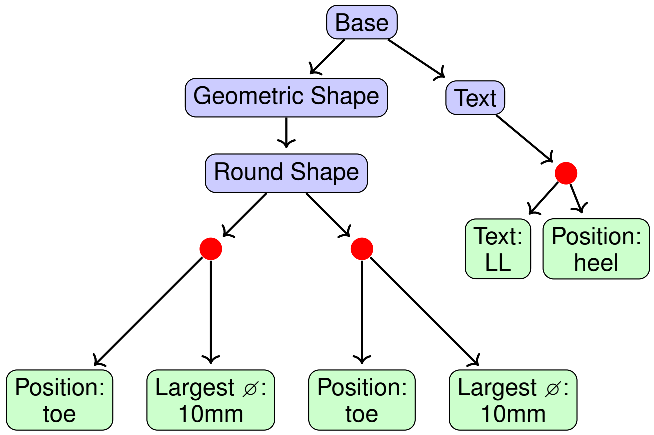

Figure 3. Example tree representing a footwear print / outsole model. The red circles represent unnamed feature nodes.

pg-gram-Distance: The pg-gram-Distance is determined akin to the Jaccard Index from set union/intersection of fixed size subtrees [13].

Tree Edit distance: The tree edit distance is a generalization of the Levenshtein string distance to trees. It counts the minimal number of required removal, insert and edit operations to transform the first argument tree into the second [14] .

We opted to use the tree edit distance as our similarity measure for several reasons. First, it can be computed completely unsupervised and requires no training. Secondly, there are quite efficient computation methods for determining the tree edit distance of ordered trees that employ dynamic programming and have quadratic time complexity in respect to the number of tree nodes [14]. Finally, most implementations allow the insert, edit and delete operations to be weighted differently, which is an advantage over the tree kernel and a necessity for our scenario. This is because a certain feature found at a footwear print should also show up in the outsole model, while an outsole model feature can be missing at the footwear print. Hence, we want a deletion operation on nodes belonging to footwear prints to be more expensive than an insertion. However, if we compare two footwear prints wich each other directly, then the deletion and insertion weights should actually be identical.

The computational complexity of the tree edit distance is polynomial for ordered and NP-hard for unordered trees. We can impose an ordering on the attribute and feature type nodes by comparing their names alphabetically. However, there is no meaningful way to compare two feature nodes other than for equality, so they are actually unordered. Thus, if we want to compare two trees, where the first argument tree contains a feature type node that is the parent of several feature nodes, we compute the ordered tree edit distance for all possible permutations of such feature nodes and take its maximum value as overall tree edit distance. Formally, this distance is given by:

$$sim_u(t_1, t_2) = \max\{sim_o(u, t_2) | u \in perm(t_1)\} \tag{1}$$

where $perm(t_1)$ is the set of all permuted trees. Consider for example a tree, in which two feature type nodes have more than one child, namely the first node two and the second one three, then the set $perm(t_1)$ consists of $2 \times 3 = 6$ elements in total.

Note that we use fixed weights for insertion and deletion operations. These weights must be specified in advance in a configuration file and can be adjusted by the forensic footwear examiner for each node type individually. For node modifications, the total weight is given by the product of a node type-specific weight, which is constant and can be adjusted in a configuration file as well, and the value of a similarity function applied on the compared nodes. For non-attributable and nominal-scaled attribute nodes, this function always assumes the value of zero for identical nodes and one otherwise. However, if the attribute values are ordinally or cardinally scaled, we would like small deviation of attribute values to be less penalized than large differences. Hence, we define the node similarity function as follows: $sim(n_1, n_2) = e^{-\frac{(val(n_1)-val(n_2))^2}{\delta}}$ where the parameter δ determines the slope or decay rate of the function and $val : Attr \rightarrow \mathbb{R}$ denotes a function that is defined on the set of all attributes $Attr$ and retrieves the attribute's current numerical value.

IV. Evaluation

In this section, we will describe the evaluation we conducted to demonstrate the usefulness of our proposed approach.

A. Goals of the evaluation

The goal of the evaluation is to get answers to the following questions:

1) How accurate is the fuzzy search?
2) Are the results reproducible with different users?
3) Does the accuracy change if user search a reference or a crime scene print

B. Description of the used test images

Kortylewski [4] published labeled images with real prints from crime scenes and corresponding references. These images were provided from different German police departments for a standardized evaluation of search algorithms. This data set allows anyone to reproduce results of published algorithms and check them for reproducibility. The images show different types of quality (c.f. image 1) and ensure that tested performances have an informative value concerning the later daily use in the field. In addition, thanks to publically available data, results from different publications can be compared. For this evaluation all 1 500 references and all the 300 prints from crime scenes provided by [4] were integrated into the testing system. To get a reasonable workload when searching for prints from crime scenes additional 350 images where introduced to the testing system. To make the images searchable, all of them were codified by the same person.

Our method was tested by a forensic scientist, experienced with the algorithm (P1), an ordinary person, already experienced on how to use our fuzzy search (P2), and another ordinary person not having any such experience (P3).

TABLE I. RANKING WITH 650 FOOTWEAR PRINTS FROM CRIME SCENES.

No.	1	3	4	5	6	8	9	10	14	15	16	18	19	23	24	27	28	29	31	33
Rank User 1	1	3	1	2	4	1	33	1	2	12	3	2	1	27	1	11	15	1	1	1
Rank User 2	3	7	1	5	1	4	1	4	1	4	2	18	1	6	12	4	12	4	1	4
Rank User 3	5	5	2	1	1	30	1	1	1	1	5	15	4	16	1	3	3	1	2	4

TABLE II. RANKING WITH 1500 FOOTWEAR PRINTS FROM CRIME SCENES.

No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Rank User 1	8	8	31	26	1	7	13	1	1	1	10	4	3	1	22	14	27	48	21	1
Rank User 2	1	17	1	11	1	15	27	11	7	12	22	2	10	2	11	25	4	6	16	3
Rank User 3	15	10	27	1	5	14	32	1	7	12	14	3	1	1	18	40	8	48	1	1

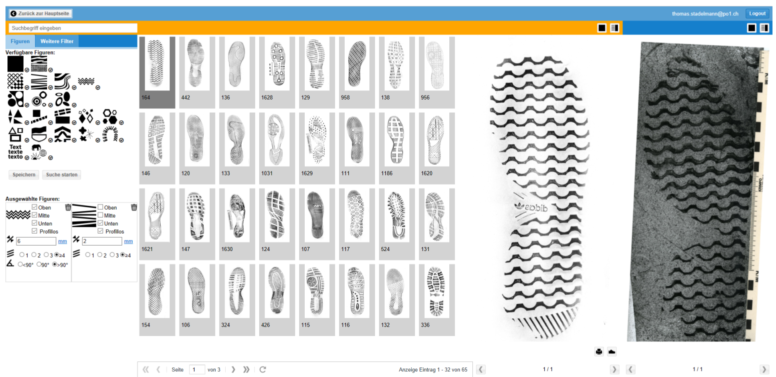

Figure 4. Screenshot of the Fast user interface.

C. Test system and test procedure

We conducted an online experiment, in which people annotated footwear print images via a browser-based web application. Once logged in, the participants first had to select a print from a crime scene and decide afterward whether they want to search for corresponding outsole models or rather for corresponding prints from other crime scenes. The workflow for processing an arbitrary footwear print image consisted of the following five steps:

1) Selecting a feature to describe the sole pattern on the image
2) Selecting attributes to specify the selected feature
3) Launch search request
4) Check results on the preview screen
5) Recording the position of the corresponding image

The participants could repeat the steps 1 to 4 as often as desired. In this way, they were able to assign multiple features and could, therefore, describe the sole pattern as precise as possible. The users were told to repeat the steps 1 to 4 until they found the corresponding images but not to codify the image as much as possible.

Our system was running on a web server with 8 vCores and 32 GB RAM. For every requested footwear print, the system retrieved the 32 most similar entries (footwear prints or outsole models) from the database and displayed them on the screen as thumbnails. The participants could click on the thumbnails to obtain a larger preview image. If the corresponding result could not be found on the first page, the participants were able to switch to the next 32 results.

All test participants received at first a short introduction of about 30 minutes. In this tutorial, they were introduced to the overall functionality of the system and the exact procedure of the test. Afterward, they were asked to independently identify the corresponding reference and prints from crime scenes for the 20 prints.

D. Results

The results for both search types (footwear print - outsole pattern and footwear print - footwear print) are presented in tables I and II (cf. Figure 5 and 6). For every request, the final position of the corresponding result is indicated. Yellow fields shoe results between position 33 and 64. This means, user preferred to scroll to the second page to get the results instead of undertaking a better coding.

For both search processes, the tables demonstrate that the difference between the rankings conducted by individual users is rather small and the system is easily applicable for both - forensic scientists and unexperienced users. In our experiment, a single user needed up to 2 hours to process all the 40 test samples. They reported to us that to the end of the experiment they were able to speed up considerably because they got meanwhile familiar to the system.

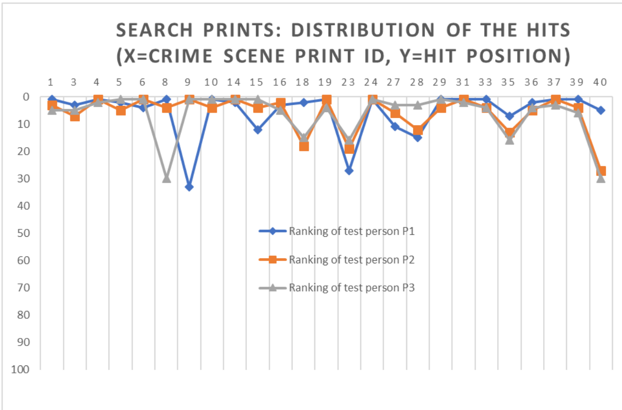

Figure 5. Ranking with 650 prints from crime scenes.

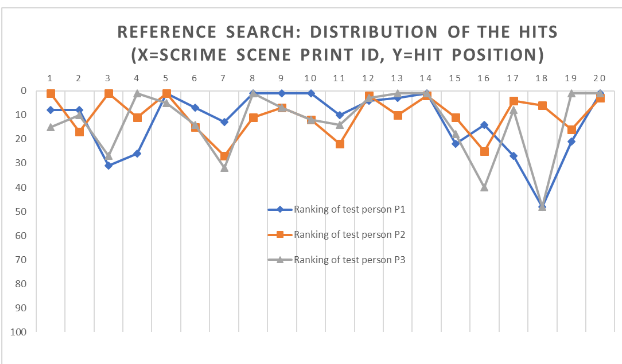

Figure 6. Ranking with 1500 prints from crime scenes.

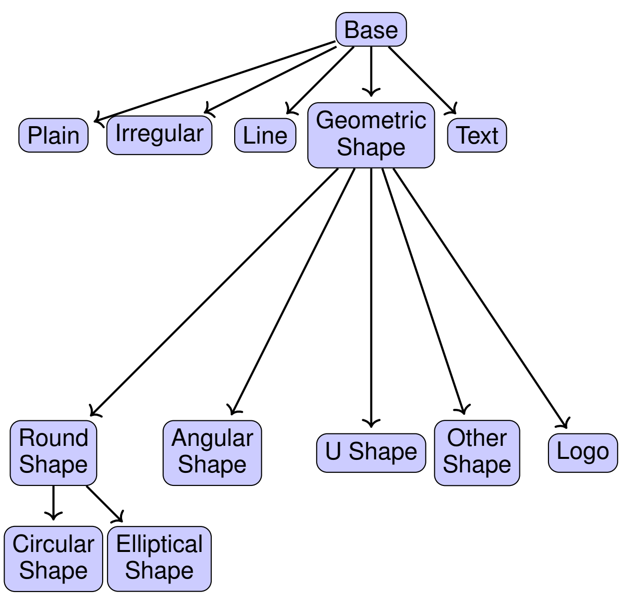

Figure 7. Feature Hierarchy

TABLE III. ALL EMPLOYED FEATURE TYPES AND ASSOCIATED ATTRIBUTES.

Base	
Attribute	Type
Position Tip	Boolean
Position Middle	Boolean
Position Heel	Boolean
Position Border Tip / Middle	Boolean
Position Border Heel	Boolean
Plain	
Attribute	Type
-	-
Irregular	
Attribute	Type
Texture Type	Enumeration (Crepe, Spots)
Spots Thickness	Floating point
Line	
Attribute	Type
Width	Floating point
Shape	Enumeration (round, straight segments)
Number of connected segments	Integer
Angle between segments	Floating point
Number of parallel lines	Integer
Distance between parallel lines	Double
Amount of crossed lines	Integer
Angle between crossed lines	Floating point
Geometric Shape	
Attribute	Type
Amount	Integer
Distance	Floating point
Round Shape	
Attribute	Type
Largest diameter	Floating point
Number concentric forms	Integer
Distance between concentric forms	Floating point
Filled	Boolean
Ring	Boolean
Ring width	Floating point
Circular Shape	
Attribute	Type
-	-
Elliptical Shape	
Attribute	Type
Shorter radius	Floating point
Angular Shape	
Attribute	Type
Number corners	Integer
Largest length	Floating point
Filled	Boolean
Regular	Boolean
U-Shape	
Attribute	Type
-	-
Other Shape	
Attribute	Value Type
Army cross	Boolean
Army rand	Boolean
Logo	
Attribute	Type
Trademark	String
Text	
Attribute	Value Type
Font size	Floating point
Text length	Floating point
Text	String

 ISBN: 978-1-61208-740-5

V. Conclusion and Future Work

We presented a novel approach for footwear print retrieval based on a hierarchical tree representation of footwear prints and outsole models consisting of features, feature types and attribute nodes. The proposed tree representation allows for establishing matches between entries of different feature/attribute orderings or levels of abstraction. As similarity measure, we opted for the tree edit distance due to its ability for incorporating weights and its polynomial runtime complexity. We applied our approach on a given set of footwear prints and compared the automatic assignments with that of human annotators. The evaluation showed the usefulness of our system and a high degree of correspondence between human and automated search results, even though we did not spend much time on parameter tuning yet. We identified possible future work in the following areas: Search process, GUI, and evaluation.

A. Search process

Our approach is highly customizable by a large number of weights, which can be adjusted separately for the individual type of modification (insertion, deletion, and replacement), node type (feature, feature type and attributes) and comparison mode (footwear prints against each other or footwear print vs. outsole model). Since a manual specification of so many parameters is quite tedious and typically involves a lot of trial and error cycles, we plan for future work to implement a genetic algorithm that adjusts them automatically.

B. GUI

Currently there is no automated check if the features and attributes, the forensic expert enters, actually matches to patterns in the footwear image. However, such a cross-check would be very beneficial but requires advanced image processing techniques. One could even go a step further and generate a first guess for the perceived features and attributes.

Besides, our system only lists similar outsole patterns but does not make any statement, whether the similarity is actually close enough so that the footwear print actually could belong to one of the identified outsole models or not. To accomplish this, one would have to derive some kind of threshold value for our similarity score that discerns the two possible outcomes *Match found* and *There is not match*.

C. Evaluation

It would be interesting to investigate, how the number of features and the accuracy of the system correlates. Finally, possible future work also includes an evaluation of other state-of-the-art systems on our dataset, which would allow for a quantitative comparison.

Acknowledgments

We thank our working collegues for their support regarding this paper and the InnoSuisse organization for funding this research.

References

[1] V. S. S. Mikkonen and P.Heinonen, “Use of footwear impressions in crime scene investigations assisted by computerized footwear collection system,” Forensic Science International, vol. 82, no. 1, 1996, pp. 67–79.

[2] A. Girod, “Computerized classification of the shoeprints of burglars' shoes,” Forensic Science International, vol. 82, no. 1, 1996, pp. 59–65.

[3] A. Girod, C. Champod, and O. Ribaux, Trace de Souliers. Lausanne, Switzerland: Presse polytechniques et universitaires romandes, 2008.

[4] A. Kortylewski, “Model-based image analysis for forensic shoe print recognition,” Ph.D. dissertation, Basel University, 2017.

[5] W. Ashley, “What shoe was that? the use of compuerized image database to assist in identification,” Forensic Science International, vol. 82, no. 1, 1996, pp. 7–20.

[6] R. Davis, “An intelligence approach to footwear marks and toolmarks,” Journal of the Forensic Science Society, vol. 21, no. 3, 1981, pp. 183–193.

[7] H. Majammaa, “Footwear databases used in police and forensic laboratories,” Information Bulletin for Shueprint/Toolmark Examiners, 2000, pp. 133–157.

[8] A. Girod, “Efficiency of computerised database of burlarsoles' standards,” Information Bulletin for Shoeprint/Toolmark Examiners, vol. 6, no. 1, 2000, pp. 125–132.

[9] S. Gross, D. Jeppesen, and C. Neumann, “The variability and significance of class characteristics in footware impressions,” Journal of Forensic Identification, vol. 63, no. 3, 2013, pp. 332–351.

[10] S. N. Srihari, “Analysis of footware impression evidence,” U.S. Department of Justice, Tech. Rep. 233981, 2011.

[11] N. Khosla and V. Venkataraman, “Building image-based shoe search using convolutional neural networks,” Stanford University, Tech. Rep. CS23 / Course Project Reports, 2015.

[12] A. Moschitti, “Efficient convolution kernels for depencency and constituent synatactic trees,” in Proceedings of the European Conference on Machine Learning, 2006, pp. 18–22.

[13] N. Augsten, M. Böhlen, and J. Gamper, “Approximate matching of hierarchical data using pq-grams,” in Proceedings of the 31st International Conference on Very Large Databases, 2005.

[14] M. Pawlik and N. Augsten, “Tree edit distance: Robust and memory-efficient,” Information Systems, vol. 56, 2016, pp. 157–173.