Andreas Daniel Hartl

Mobile Interactive Document Verification

DOCTORAL THESIS

to achieve the university degree of
Doktor der technischen Wissenschaften

submitted to

Graz University of Technology

Thesis Supervisor
Prof. Dr. Dieter Schmalstieg
Institute for Computer Graphics and Vision

Referee
Prof. Dr. Hideo Saito
Department of Information and Computer Science
Keio University, Kanagawa, Japan

Graz, Austria, July 2015

To my Parents

The passport is a human being's noblest part. It comes into being much less simply than people themselves do. A human being can come into the world anywhere, in the most careless way; but a passport, never. For that reason it is recognized when it is good, whereas a human being can be very, very good yet go unrecognized.

Bertolt Brecht

Abstract

Document inspection requires detailed knowledge, but it is carried out by trained and untrained individuals. Although the process can be automated by dedicated machinery, such devices are not always available and the correct interpretation of results requires training. Due to the widespread use of smartphones, it is interesting to investigate their usefulness regarding document inspection. However, this poses unique challenges due to a large and diverse corpus of documents, unexpected user behavior and limited resources. In particular, we investigate the usefulness of Handheld Augmented Reality setups as semi-automatic tools for document inspection. We first investigate the detection and classification of documents for making them accessible in a mobile setting. For this task we employ an efficient approach for the detection of the document region, which allows to process rectified images in a client-side solution for mobile visual search. This improves both tracking and classification performance over using full-frame images and gives instant results. We show that the client-side engine compares favorably to a commercial solution and also delivers reasonable performance with document images. We further aim to exploit handheld setups beyond being document information systems by extracting textual information from documents. We propose an efficient solution for reading machine-readable zones using the built-in camera of a smartphone without imposing strict limitations on the viewpoint, which standard applications do. The extracted data can be instantly used for verification or for querying additional information. We also contribute a large set of synthetically generated data for further research. With the goal to support the inspection of optically variable devices such as holograms, we initially show the feasibility of detecting, recording and matching them in a mobile setting. For reasons of task complexity, the user should be guided throughout the image capture process, which we tackle by presenting several user interfaces for view alignment and constrained navigation. We finally show that a parametrization based on typical user behavior considerably reduces the temporal effort, while providing slightly better decisions than untrained users.

Kurzfassung

Die Prüfung von Dokumenten wird von Personen mit unterschiedlichem Wissensstand durchgeführt. Spezielle Prüfgeräte sind nicht immer verfügbar und die korrekte Interpretation der Ergebnisse benötigt entsprechende Kenntnisse. Durch die große Verbreitung von Mobiltelefonen steht grundsätzlich ein weiteres Prüfwerkzeug zur Verfügung. Besondere Herausforderungen bestehen im großen und vielfältigen Bestand von Dokumenten, im weitgehend beliebigen Verhalten der Nutzer sowie in den eingeschränkten Ressourcen von mobilen Geräten. In dieser Arbeit beschäftigen wir uns im Speziellen mit dem Einsatz von Handheld Augmented Reality zur semi-automatischen Prüfung von Dokumenten. Zunächst wird die Detektion und Klassifikation von Dokumenten betrachtet. Wir stellen dafür einen effizienten Ansatz zur Detektion des Dokumentbereiches vor. Damit kann ein entzerrtes Bild ermittelt werden, welches dann in einem lokalen Ansatz für Mobile Visual Search verarbeitet wird. Diese Vorgangsweise verbessert sowohl das Tracking als auch die Klassifikation. Wir zeigen außerdem, dass der lokale Ansatz eine vergleichbare Erkennungsleistung wie eine kommerzielle Lösung bietet, aber deutlich weniger Zeit benötigt. Die Eignung zur Klassifikation von Dokumenten wird in einer Evaluierung gezeigt. Wir stellen außerdem eine effiziente Lösung zum Auslesen der maschinenlesbaren Zone vor, wobei nur die eingebaute Kamera eines mobilen Gerätes verwendet wird. Damit können im Gegensatz zu aktuellen Lösungen auch rotierte und perspektivisch verzerrte Dokumente erfasst werden. Die extrahierten Daten können unmittelbar zur weiteren Prüfung oder zur Abfrage von Zusatzinformationen dienen. Es wurde außerdem ein großer Bestand von Daten erzeugt, welcher als Grundlage zur weiteren Forschung dienen kann. Mit dem Ziel, die mobile Prüfung von optisch variablen Elemente wie Hologrammen zu unterstützen, wird zunächst die Machbarkeit von Detektion, Aufnahme und Abgleich gezeigt. Aufgrund der besonderen Anforderungen dieser Anordnung, muss der Benutzer zur Aufnahme angeleitet werden. Dies wird durch spezielle Benutzerschnittstellen zur gezielten Ausrichtung und zur Bewegung in einem beschränkten Navigationsbereich erreicht. Mit einer Parametrierung basierend auf dem typischen Verhalten von Nutzern kann der Zeitaufwand deutlich reduziert werden. Das System übertrifft letztlich die Entscheidungen von Laien bezüglich der Echtheit der verwendeten Hologramme.

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

The text document uploaded to TUGRAZonline is identical to the presented doctoral thesis.

Place	Date	Signature

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Dissertation identisch.

Ort	Datum	Unterschrift

Acknowledgments

First, I would like to thank Prof. Gerhard Reitmayr for awakening my interest in document verification using mobile devices through an initial feasibility study, and for supervising me during parts of this thesis along with the associated research project. Thanks go to Prof. Dieter Schmalstieg for supervising me during all subsequent scientific efforts until the completion of this thesis and to Prof. Hideo Saito for reviewing my work. Bundesdruckerei GmbH and in particular Olaf Dressel shall also be thanked here for their sustaining collaboration throughout this thesis.

Among my colleagues from the Institute of Computer Graphics and Vision, Clemens Arth deserves special mentioning for his continuous support, motivation and ambition to generate and follow new ideas. I would also like to thank Jens Grubert, Christian Reinbacher and Lukas Gruber for a very fruitful collaboration on various scientific efforts. Thanks also go to Philipp Fleck and Peter Gigler, who I supervised in student projects connected to the scope of this thesis.

Finally, I would like to thank my parents for showing me the importance of education and for supporting me. Then, my brother Stefan as well as my uncles Wolfgang and Günter deserve mentioning, since each of them supported my interest in technology in their unique way and cheered me up during critical stages of this work. Special thanks go to my dear girlfriend Ganna for her understanding during the final phase of this thesis.

Contents

1 Introduction

Contents

1.1 Mobile Augmented Reality

Augmented reality (AR) [22] aims to enrich the real world with digital information in order to generate value for a human operator. The first prototype (The Sword of Damocles) was realized in 1968 by Sutherland [155] as a head-mounted display rendering wireframe models according to input from a mechanical or ultrasonic head position sensor. However, a widely accepted definition of the term AR did not appear until the 1990s. Milgram et al. [113] define a Reality-Virtuality Continuum and position AR into the space between reality and virtual reality. A definition given by Azuma [7] provides more detail by stating that AR applications

- combine the real and the virtual,
- are interactive in real time and
- register information in the real world in 3D.

We are interested in setups which are at least partly aware of their surroundings and are able to extend objects or parts of the environment with purposeful information. These are

dynamic setups, where the display of information can adapt according to the viewpoint in order to allow a seamless experience for the user. In the past, AR systems often required dedicated, powerful and, thus, non-portable hardware, which in turn limited its distribution. This has changed with the world-wide adoption of reasonably powerful mobile devices with built-in cameras.

According to recent estimates, one quarter of the global population will be using smartphones in 2015[1]. Nowadays mobile devices employ several computing cores, programmable GPUs and high-resolution cameras as well as screens. Besides, there is a wide range of sensing and communication facilities included. Mobile performance, although still limited in extent for reasons of the form factor, manages to exceed performance of desktop machines from a few years ago. Consequently, there is a trend towards substituting desktop setups and laptops with smartphones and tablets for certain productive tasks, while overcoming limited storage capabilities and locality with cloud solutions. With such a wide distribution of devices at hand, new opportunities and application scenarios evolve, which also bring up new challenges in computer vision (CV), visualization and interaction. From the early steps of marker-based tracking, which was later extended to natural features, there is a trend towards the reconstruction of the environment or individual objects, which allows the adaptation of an application to the current setting. Screens, although offering a reasonably high resolution, are still small in physical size and require sophisticated strategies for visualizing information. Interaction is no longer limited to buttons or the screen. Instead, devices and even the environment serve as tangible user interfaces, where the basis must be provided by robust, efficient and scalable CV algorithms.

While many scenarios for Mobile AR target navigation or entertainment, we propose the use such setups as a tool for the inspection of documents. This is motivated by the fact that mobile devices are readily available, but dedicated tools and, in particular, knowledge for the inspection of documents by the public are not. Besides, mobile services for document authentication[2] are currently emerging, which underlines the relevance for scientific treatment of this topic. In the following, a brief introduction into the field of document inspection will be provided, including considerations on the opportunities for mobile devices in this context. Then, the goals for this work are defined along with a description of the scientific contributions of this thesis.

1.2 Document Inspection

The purpose of document inspection is to reason about the validity of a document by examination of document properties, including security elements. Documents of interest are machine-readable travel documents, like passports, identification cards and visas, but also checks, vouchers and banknotes. The outcome of document inspection may affect the

[1] `http://www.emarketer.com/Article/2-Billion-Consumers-Worldwide-Smartphones-by-2016/1011694`

[2] `https://www.jumio.com`

ability of a person to use a document for payment up to the point of gaining access to restricted facilities.

According to a press release by the European Central Bank [41], the number of counterfeit Euro banknotes withdrawn in the second half of 2014 was about 507,000. This example shows that everyone should be concerned about document counterfeiting and that there is obviously an information deficit in the public on how to prevent fraud in everyday cash payment.

In the remainder of this work, specific terms for document inspection will be used, which are now defined:

Counterfeit Document: Unauthorized reproduction of a full document.
Forged/Tampered Document: Unauthorized alteration of an original.
Pseudo Document: Fantasy/fictitious or camouflage document without legal value.

1.2.1 Security Features

Valuable documents contain various security features, with the goal to prevent forgery or counterfeiting. Corresponding features should be cheap in mass production, but require expert knowledge and specialized machinery for reproduction. In general, advanced security features must not allow a cheap shortcut in their inspection. Still, human inspection must be possible with manageable effort and training. Features should be designed to allow for automatic checks by appropriate machinery in order to save time and to avoid human error. On the scale of the document, there should be features which also allow basic checking for blind or partially sighted people. While, in the following, a short overview of popular security features will be given, the interested reader is referred to the work of van Renesse [169].

Overview: There are a large number of security features currently in use. They can be incorporated into the substrate of a security document, printed on top, or added as a separate element, sometimes as an additional foil containing several features over the entire document. Security features can have haptic properties, which must be checked by touching. Examples are the type of substrate, imprints, intaglio (high pressure) printing, perforations or watermarks (see Figure 1.1). The latter are more commonly checked by viewing them with light from behind, so these are also optical security elements.

Similar, there are registers, which consist of parts printed on the front- and backside of a document, but unite to a full element when held against a light source. There is also microprint, which is a very high resolution printing that cannot be copied easily. Similarly, guilloches consist of very fine geometric structures, which are often used to impede the alteration of personal information present on a document.

Optically variable devices (OVD) have distinct visual properties, which change con-

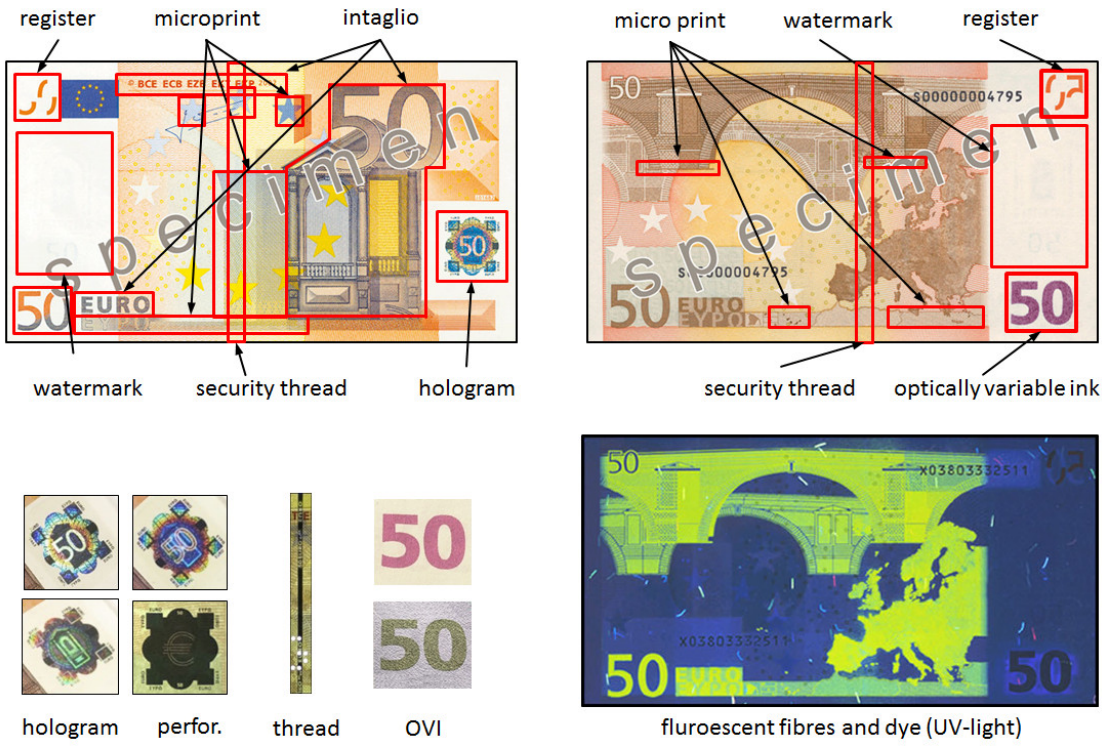

Figure 1.1: Subset of the security features present on a 50 Euro banknote (top row) with selected details for optically variable features (bottom left).

siderably when varying the viewing angle or the position of incident light sources. A well known feature used on banknotes is optically variable ink, which changes its brightness or tone depending on the viewing angle. Similarly, a stripe (thread) can be incorporated on top or inside of the substrate. Then, there are security holograms, which show one or more distinct patterns, depending on the viewing angle and sources of illumination, exploiting various physical phenomena (diffraction, interference). Often, so-called rainbow holograms are encountered, which can be viewed by using white light, letting the object appear in all spectral colors. Depending on the number of layers, 2D and 3D images or motion sequence (stereogram, kinegram) can be shown. Although they generally show iridescent colors, a more natural appearance can be achieved (e.g., true color hologram). Changeable (CLI) or multiple (MLI) laser images contain various pieces of information written by a laser into the substrate, where, depending on the viewing angle, only a single one can be seen at a time. Note that throughout this thesis, we use the term hologram interchangeably with the terms OVD or view-dependent element.

There are other features like thermochromatic (color change depending on temperature) or magnetic ink. Windows can also be incorporated, which can act as filtering devices to visualize other security feature present on a document (self-authenticating). However, the most advanced element is only relevant when it does not fall victim to the

complacency of the person in charge.

For inspection by dedicated machinery, often fluorescent dyes and substrates are used, which make the document change appearance, when viewed under infrared or ultra-violet light. Special machine-readable zones (MRZ) are used on certain identity documents, which contain the most important information about the document and the corresponding person including checksums. They allow for efficient and accurate readings by dedicated machinery. Following a suggestion by the International Civil Aviation Organization (ICAO) in 2003, biometric passports were introduced (ePassport). They store personal information and, possibly, also biometric features such as fingerprints or a facial image on a microchip, where the latter can only be obtained via a secure communications channel employing near-field communication (NFC).

Threats: A list of main threats for the security of travel documents can be found in an ICAO document [74]. They state that the entire document can be counterfeited and that elements or pages are substituted, deleted or altered. Fraudulent documents may be constructed using material from legitimate documents for by using illegally acquired genuine document blanks. In order to prevent copying, when digital reproduction equipment is at hand, the use of optically variable features with appropriate integration into the document is suggested. Although they cannot be copied by photo-mechanical means, OVDs and, in particular, holograms can be attacked by re-origination (complete remake), performing optical or mechanical replication, including the use of substitutes. In case of Euro banknotes, a special ring pattern (EURion constellation) is printed onto the document during production. Upon its detection, off-the-shelf digital copying equipment will not allow duplication. Sometimes, fake documents are artificially aged in order to make imperfections of features appear less obvious. A microchip present on a fake document can be deliberately destroyed to prevent comparisons of visible and stored information in the chip. For travel documents, there seems to be a shift from fraudulent alteration of documents towards identity fraud, where look-a-likes try to pass personal checks using a genuine document from another individual [33]. In this case, it is no longer sufficient to mainly check the document itself, but to put equal effort in the recognition of the person.

1.2.2 Inspection and Tools

A common way for checking a document mainly for untrained individuals is the feel-look-tilt test. This is often carried out with banknotes[3]. The examiner is required to touch certain parts of the document in order to feel the structure of the surface and to carry out visual inspection by looking at the document and comparing the appearance with reference material often found in a manual. This may include certain movements of the document, in order to trigger a visual effect relevant to inspection. In case of ID-documents, an additional check for correspondence of the data on the document and its owner is required

[3] `http://www.ecb.europa.eu/securityfeatures`

Figure 1.2: Exemplary tools for document inspection (left to right, top to bottom): Banknote Pen (`http://www.amazon.co.uk`), Manual (`https://www.ecb.europa.eu`), Doculus Lumus (Magnifying Glass, LEDs (White, UV), NFC Reader - `http://www.doculuslumus.com`), handheld and Passport Readers (`http://www.access-is.com`), Smartphone

in order to catch impostors. Although, until recently, this was limited to an examination of the document itself, an additional security layer has been added by the introduction of a micro-chip containing biometric data (e.g., facial image, fingerprint or iris scan). However, an unreadable chip does not automatically render such an ID-document invalid. Consequently, checking documents still requires thorough inspection by human beings or special machinery.

More formally, the inspection of documents is often divided into three different classes [169]. They differ in the type of tools and the amount of additional knowledge (training) involved throughout the inspection process. The general public typically carries out first-line inspection, which does not involve special training or the use of additional tools. Typically, watermarks, intaglio printing, security threads, registered printing and view-dependent elements such as holograms or optically variable ink are examined. Special pens (check of paper type) or reference manuals are used for checking banknotes at this level (see Figure 1.2). In second-line inspection, trained people check documents using special tools, which often provide an automatic decision on validity. In this case, features such as luminescent printing, magnetic inks, or bar codes are used. This level also includes checks performed in automatic teller machines. Forensic experts carry out third-level inspections using even more sophisticated equipment (e.g., spectrometers, microscopes, infrared cameras and chemical indicators). This often happens in laboratories or a dedicated inspection facility. In contrast to the other levels of inspection, this may be destructive to the document in question.

According to van Renesse [169], it is not possible to replicate every part of a security document faithfully, and not all security features present on a document are known to the public. Comparisons with the naked eye are considered sufficient for checking, but this requires detailed information about the document. Consequently, he states that there is a tendency to reject documents instead of learning how to inspect them properly. In case of holograms, first-line inspection is based on printed guides or digital manuals. Often being issued by public authorities, they usually show distinct patterns visible within the hologram area. However, they often lack an indication on the viewing direction and do not specify requirements on the lighting conditions. Consequently, the inspection may be tedious for the untrained user. In the absence of specific information, holograms tend to be inspected just by looking for changes in appearance or the pure presence of rainbow colors, which has no particular value with respect to security [169].

1.2.3 Opportunities for Mobile Devices

Gariup and Soederlind [51] compared the performance of experts and automatic systems for document verification in a study. They found that automatic inspection was slower and performed less well. Closed studies reveal that during inspection, individuals tend to overly rely on automated document readers instead of following their instinct. Besides, the chance of misinterpretation of results may constitute a vulnerability and, consequently, thorough training is required. They conclude that human factors are important and that automation may become a risk. This is further backed up by a Gschwandtner et al. [56], who show that automatic border control systems can be tricked by using an active display along with some sensing hardware.

It must be noted that using a mobile device would correspond to second-level inspection, because it is an additional tool. However, in certain cases the process requires interaction with a human operator and thus cannot be used to assess documents fully automatically. Besides, due to the wide user-base of mobile devices, it can also be used by lay people. With the aforementioned considerations on human behavior with fully automatic inspection, it is very reasonable to propose a semi-automatic setup involving an off-the-shelf mobile device.

When using off-the-shelf devices for document inspection, only the built-in features of the device are available. This includes the front and back camera, usually with automatic focus, in most cases also a white LED flashlight and the capability to perform near-field communication (NFC). The use of UV sources is not common, and even using an external one would not be useful, since phone cameras nowadays employ UV filters. Not considering the built-in microchip, such devices can only serve as advanced magnifying glasses, if no specific applications targeted towards document inspection are available. Given the appropriate software, they could serve as advanced information systems, gain the capability to read and match information from a document or even serve as an advanced tool for the inspection of specific elements on a document.

Information System: Finding the correct reference information for a given document is an important requirement prior to the actual inspection process. This can be achieved by requiring a manual lookup of document properties (e.g., country, document type, year of issue etc.) using a manual, or by entering such information within an application for mobile devices. However, this could also be achieved by carrying out the analysis of one or more images taken by the built-in camera of a mobile device. If, at this point, the relevant set of documents is known to the system, pseudo-documents will be immediately rejected. For verification of security features, there is still a mapping required from the information shown on the screen to the document and vice versa. By tracking the document, the position of security features and, possibly, their appearance can be augmented in the right spot, which can facilitate the mapping of information by the operator. In the end, this leads to the design of an advanced AR information system for document inspection.

As obtained from an initial study we conducted, people mainly focus on holograms, faces and textual information (see Appendix B). Surprisingly, the latter seems to be less important for a quick check and, in the absence of tools, the MRZ cannot be thoroughly examined at all, since there are checksums included.

Extraction and Matching: However, it is desirable to provide further support by automatic extraction and matching of information [50]. This could speed up the process and increase robustness by removing human bias. In particular, this leads to the detection and extraction of face images, text and all kinds of special patterns. With the goal to allow fast and efficient queries about the person in question, a machine-readable zone has been put on documents starting around 1980. Such information can be detected and extracted using off-the-shelf mobile devices. However, special care must be taken in order to allow for fast and robust reading despite changes regarding the viewpoint and image capture conditions, which are characteristic for mobile scenarios.

View-Dependent Elements: Holograms are considered highly fraud-resistant. Van Renesse states that, according to the International Hologram Manufacturers Association (IHMA), there has been no case of a well-designed authentication hologram being copied accurately [169]. Therefore, the development of tools and algorithms for the mobile detection and verification of such elements deserves interest in research. In particular, these are interesting features for inspection in mobile scenarios, since checking such features requires a movement of the document in order to reproduce the desired appearance for comparison, which is the case in typical Mobile AR setups. When tracking documents containing these elements, the image capture conditions can be monitored. This could pave the way for semi-automatic inspection systems for holograms and similar elements. However, such a setup poses unique challenges regarding user guidance, visualization and the selection of relevant reference information.

1.3 Problem Statement

The following considerations target the inspection of documents using off-the-shelf mobile devices in the context of AR. This involves the active support of the verification process by automatic image capture, extraction of relevant information and automatic matching with reference information.

A possible application scenario is the inspection of documents by small shops or individuals, which do not carry out this task on a regular basis and, thus, are not willing to spend money on dedicated machinery or carry it around in their pocket permanently. Mobile AR setups for document inspection could also be useful for educational purposes and for raising awareness of the public regarding newly introduced or updated documents.

Vision: An ideal Mobile AR system for document inspection would perform an automatic classification of the document, taking away the burden of manual document selection from the user. All steps involving identity documents are carried out directly on the mobile device for reasons of protecting sensitive data. This requires a classification and tracking approach that is largely independent of personal information shown on the document. Then, relevant information can be immediately overlaid onto the document and visually marked according to its importance throughout the process. Identity information is read seamlessly from the document and used for querying supplementary information about an individual from a remote databases connected over a secure communication channel. As, typically, the operator has both hands employed with holding the device and the document, the selection of appropriate information is carried out by changing the pose of the document or the device. During slight pose changes of the user, the behavior of view-dependent elements can be monitored and matched against reference information. Suitable hints and visual feedback given by the system help the user to efficiently capture all the required information. The system finally provides its verification result and provides insights to the user on the basis of a particular decision.

Goals: The goals of this thesis originate from the desire to exhibit Mobile AR setups for document inspection beyond being a medium for in situ information presentation. They can be summarized as follows:

- Enable the mobile *detection and tracking* of arbitrarily personalized documents.
- Investigate how off-the-shelf mobile devices can serve as *verification tools* by reading textual information present on documents and, in particular, supporting the mobile *verification of view-dependent elements* such as holograms.

It must be noted that we do not address face recognition for the purpose of inspecting ID-documents in this thesis. The reason is that this task deserves extensive treatment, which is out of scope here. Besides, face recognition and verification with the image on

a document requires taking one or more facial images of a person, which would put the focus away from the document, thus, interrupt the work-flow and, possibly, cause a certain degree of indisposition.

1.4 Challenges and Strategies

The aforementioned goals give rise to specific challenges that need to be addressed throughout the thesis. First, there are lots of different document types and variants floating around, for which little or no reference data is accessible. In many cases, this hinders the design of specific approaches, which are otherwise deemed well suited. If applicable, we perform a detailed evaluation, often directly on mobile devices.

ID-documents are personalized regarding the owner, which affects their appearance and recognition performance dramatically, including the ability to track them from a given template. We split the recognition and tracking task and first localize the document in order to build a tracking-target on the fly. However, some state-of-the art frameworks for tracking require each template to be sent over to a server. This is not feasible for reasons of protecting personal data. So, all processing involving ID-data on documents must be carried out directly on the mobile device. The limited resources available on mobile devices clearly conflict with desirable properties from the perspective of the user, such as instant processing and feedback.

The user should not be rigidly constrained in his or her movements. This is an important requirement for automatic extraction or matching of information. In order to tackle these issues, we exploit prior information about the task at hand, whenever possible, without entirely sacrificing the general applicability of the approach.

View-dependent elements such as holograms pose several unique challenges such as major changes in appearance, depending on the viewing direction and the nature of light sources in the environment. We use a light source fixed to the camera in order to reduce task complexity and allow the application on off-the-shelf mobile hardware.

Guiding the user within the small workspaces involving document and device while conducting a task pressed for time, is not obvious at all. We design and evaluate several approaches in an iterative design process, providing instant feedback on spatial position and completion status, with a presentation of results that is suitable for visual inspection by the user.

1.5 Contribution and Results

The major parts of this thesis are based on publications which were authored by myself. In particular, the following list describes the mapping of these papers to the corresponding chapters.

- Chapter 3 is based on the papers:

A. Hartl and G. Reitmayr. Rectangular target extraction for mobile augmented reality applications. In International Conference on Pattern Recognition (ICPR), 2012

A. Hartl, D. Schmalstieg and G. Reitmayr. Client-side mobile visual search. In International Conference on Computer Vision Theory and Applications (VISAPP), 2014

This chapter describes basic building blocks for document inspection using Mobile AR through document detection and tracking. We propose to extract a suitable template directly from the live-video-stream provided by the built-in camera of mobile devices. So, tracking the document can take place immediately, and the task of determining the actual class of the document is treated as a separate step. This is carried out on the obtained template by a custom client-side solution for visual search, avoiding any communication of personal image data.

- Chapter 4 is based on the paper:
  A. Hartl, C. Arth and D. Schmalstieg. Real-time detection and recognition of machine-readable zones with mobile devices. In International Conference on Computer Vision Theory and Applications (VISAPP), 2015

  This chapter describes a mobile approach for efficient detection and extraction of machine-readable zones using off-the-shelf mobile devices. It does not require strict alignment with an orthogonal viewing direction and allows continuous feedback for the user. This provides a fast and cost-effective way of reading the MRZ data for querying additional information without the need for specialized machinery. We also contribute a large database of synthetic MRZ data, covering a broad range of diverse settings, backgrounds and view points for evaluation.

- Chapter 5 is based on the papers:
  A. Hartl, C. Arth and D. Schmalstieg. Ar-based hologram detection on security documents using a mobile phone. In International Symposium on Visual Computing (ISVC), 2014

  A. Hartl, J. Grubert, D. Schmalstieg and G. Reitmayr. Mobile interactive hologram verification. In International Symposium on Mixed and Augmented Reality (ISMAR), 2013

  In this part of the thesis, we investigate the feasibility of hologram detection and verification on off-the-shelf mobile devices. In particular, we show that it is possible to detect holograms on documents without prior information on the location or appearance of these elements. Then, an approach for repeatable capture of patterns visible on holograms is presented and evaluated. The obtained

results indicate that the mobile recording and verification of such information is feasible, but the complexity of the task demands suitable user guidance.

- Chapter 6 is based on the aforementioned paper including follow-up work:
  A. Hartl, J. Grubert, C. Reinbacher, C. Arth and D. Schmalstieg. Mobile user interfaces for efficient verification of holograms. In Virtual Reality (VR), 2015

  This chapter is mainly devoted to user interfaces for mobile hologram verification and the evaluation of prototypes. First, an initial approach for view alignment is presented an evaluated within a user study. The obtained results indicate that the mobile recording and verification of holograms is feasible, but time-consuming. Building on these results, the efficiency of mobile hologram verification is improved by employing special task-oriented user interfaces along with automatic image capture and matching. After determination of the most efficient user interface for this task through a user study, an alternative parametrization is proposed. In an additional study, this setup is shown to further decrease checking time and also to improve the robustness of mobile hologram verification up to a level, which is better than decisions made by lay people.

1.6 Collaboration Statement

The aforementioned papers involved collaborations with several people from the Institute of Computer Graphics and Vision at Graz University of Technology. Clemens Arth was involved in creating new research ideas and gave early feedback regarding most of the approaches crafted throughout this thesis. He also contributed as a co-author, constantly encouraging to push forward paper drafting and submission. Gerhard Reitmayr supervised part of the thesis and also contributed an implementation of a tracker based on natural features. Jens Grubert was involved in the design and evaluation of user studies regarding hologram verification. Christian Reinbacher supported the creation of a robot setup for recording reference data of holograms and also contributed by controlling the robot arm.

1.7 Thesis Outline

The context for the scientific work conducted in this thesis is built by consideration of related work in Chapter 2. The basis for Mobile AR-based document inspection of documents with arbitrary personalization is laid in Chapter 3. An approach for reading machine-readable zones with mobile devices is proposed in Chapter 4. The feasibility of mobile hologram detection and verification is investigated in Chapter 5 and several user interfaces for hologram verification are presented and evaluated in Chapter 6. Finally, a conclusion along with an outlook on future work is provided in Chapter 7.

2 Related Work

Contents

Document inspection with off-the-shelf mobile devices takes place in a small workspace directly in front of the user. The demand to present relevant information for inspection calls for a robust and scalable approach for determining the class of an unknown document. Since OVDs can be involved, information about the relative position and orientation of the document regarding the image acquisition device is required. In an unconstrained setup, it is very reasonable to guide the user throughout the process. Due to these considerations, mobile document inspection can benefit from AR.

In the following, we consider relevant developments in Mobile and Handheld AR. We provide an overview of document inspection by means of CV, focusing on banknotes, ID-documents as well as OVDs. We identify important topics for research on document inspection with Mobile AR. First, document detection and classification are treated in the context of large-scale setups suitable for client-side processing. Then, the extraction of textual information from documents is considered. Finally, visualization and user guidance are treated, which, along with registration, are critical regarding practical application.

2.1 Towards Handheld Augmented Reality

Starting with Sutherland's initial prototype in 1968 [155], it took until the 1990s to establish a widely accepted definition of AR [113], [7]. The demand to combine the real and the

virtual world requires an accurate and continuous registration of information (pose tracking). This, in turn, allows the creation of interactive applications working in real-time, aiming to generate a benefit for the user. In the context of this thesis, pose tracking using images captured by a camera deserves a more detailed description, along with a historic outline of Handheld AR, which also illustrates current directions of research in the field.

Pose Tracking: Azuma states that pose tracking should take place in real time (i.e., 30 FPS), be stable (e.g., free of jitter), and the accuracy should be in the order of a few mm regarding position and fraction of a degree for orientation [7]. For most applications, this involves the estimation of six degrees of freedom (e.g., translation, rotation). With this information, the relation of the observing camera and a scene can be described (extrinsic parameters [65]). Together with the intrinsic parameters of the camera (principal point, focal length) determined in an off-line calibration phase, a view frustum is defined for visualization of information with respect to the viewing position and orientation of the observer. Image distortion caused by the camera is usually modeled through radial and tangential distortion and must be considered during estimation, if high accuracy is required.

Obtaining such information is a complex task and has been studied extensively in literature. In the beginning, various sensors were used for tracking along with a prior on location and orientation. These setups often suffered from inaccuracies and drift. Consequently, visual tracking was added in order to compensate for these effects. However, this comes at the cost of considerably increased computational effort, in particular for unmodified or unknown scenes.

2.1.1 Initial Setups using Sensors

Efforts to use compact devices for AR initially required stationary setups for information overlay or tracking. An example is the *Chameleon* system presented by Fitzmaurice in 1993 [47], which displays spatially situated information on a map. Employing magnetic tracking of a handheld device, graphics are rendered on a workstation, recorded via a camera and transferred to the device for display.

Advancements in mobile computing power and the availability of the Global Positioning System (GPS) for coarse registration enabled the creation of backpack systems (Mobile AR), resulting in several outdoor scenarios. In 1994, Loomis et al. [104] presented an outdoor navigation system for the visually impaired. It uses differential GPS and a head-worn compass connected to a notebook employing data from a Geographic Information System. This generates an *acoustic virtual display*, which can give audio hints at interesting locations. The first Mobile Augmented Reality System was presented by Steve Feiner et al. [45] in 1997. Their *Touring Machine* uses a see-through head-mounted display (HMD) with an orientation tracker. Other components such as differential GPS and a digital wireless radio for web access along with a notebook are put into a backpack. The system

contains also a handheld computer with a stylus and a touch interface. The application scenario is an interactive campus guide, overlaying label information onto buildings, which can be selected and navigated to, along with the possibility to get additional information from a server running on the handheld computer. They reported that, due to tracking inaccuracies, it was only possible to label an entire building instead of a particular part. In 1998, Thomas et al. [160] presented a mobile system using a HMD, GPS and a wearable computer. Initially being used for navigation purposes, it also added an electronic compass. Höllerer et al. [72] presented a system for experiencing hypermedia content in their associated location, along with a mobile campus guide, in 1999, employing for the first time GPS and inertial sensors for tracking.

A typical application using built-in sensors of current mobile devices are AR browsers (e.g., Wikitude[1], Layar[2]). They augment the environment with digital information from a remote database, according to the associated geographical location or object. Initially suffering from the the same issues in accuracy and usability as reported with early custom AR setups, they have recently added visual tracking along with image recognition capabilities.

2.1.2 Visual Tracking

As mobile setups were gaining image capturing capabilities, visual tracking became an interesting topic for the AR community. Initially, visual tracking was realized by adding artificial markers to a scene and measuring their position by evaluation of the live-feed obtained from the camera. In case of limited computational capabilities, this task can be outsourced to a server, if the corresponding latency is acceptable for the application. In 1995 Jun Rekimoto and Katashi Nagao proposed *NaviCam* [140]. They employ a mobile display tethered to a workstation, which uses a camera for optical tracking of color coded-markers. Context-sensitive information is overlaid directly on the screen of the mobile device. In 1999, Kato presented an open-source framework for tracking matrix markers in real-time (ARToolKit) [83]. It subsequently became very popular within the AR community. In 2000, Thomas et al. [159] presented an extension to the popular game Quake. Their setup consisted of GPS, an electronic compass and, most notably, vision-based tracking using markers. This mobile backpack setup allows to control the game character by movements of the user in the real world and the composition is displayed in a HMD. In 2001, Reitmayr and Schmalstieg presented a system for mobile collaborative AR [139]. They used a freely configurable tracking setup, fusing information from arbitrary sensors in order to allow the manipulation of objects in the near field. Several users can collaborate in this shared space and even join an on-going session.

In 2003, Wagner et al. presented the first AR system with stand-alone marker tracking running on a handheld device and coined the term Handheld Augmented Reality [175].

[1] `https://www.wikitude.com`
[2] `https://www.layar.com`

They used this setup as the basis for an indoor navigation application. Optionally, a server connection for offloading the tracking task from the mobile device can be used, which is transparent for the application. In the same year, the first commercially available AR game (*Mozzies* or *Mosquito Hunt*) was delivered with the Siemens SX1 phone. Mosquitoes are superimposed over the live video feed. The goal is to shoot at them using a cross hair controlled by movements of the device detected through analysis of the video feed. In 2004, Möhring demonstrated a system for detecting and tracking 3D markers on a phone, providing see-through AR without requiring a calibrated camera [119]. In 2006 Reitmayr and Drummond [138] presented a model-based tracking system for outdoor AR in urban environments, which allowed accurate overlays on a handheld device. They combined an edge-based tracker (accurate localization) with a gyroscope (coping with fast motions), along with measurements of gravity and the magnetic field for avoiding drift. In case of tracking failure, an automatic re-initialization procedure was started, which used previously stored reference frames. In 2008, Wagner et al. [174] presented the first implementation of natural feature tracking (NFT) in real-time on mobile phones. Consequently, no artificial parts (e.g., marker) need to be added to the scene. They rely on heavily optimized local features in order to cope with the limited capabilities of mobile device. In 2011, Kurz and Benhimane [91] investigated the use of gravity information for improving tracking and also augmentation in Handheld AR.

Arth et al. created several works for wide-area localization on mobile phones, using off-line reconstruction (2009)[5], only GPS-tagged images (2012)[4] or a client-side SLAM setup (2014)[171]. These approaches allow a mobile device to globally determine its position in large environments and instantly perform pose tracking.

Reconstruction and Tracking of the Environment: In 2007, Klein and Murray [85] laid the basis for AR in partially unknown environments with a system for parallel mapping and tracking (PTAM) from a monocular camera, which can automatically discover its surroundings while tracking. This was later adapted to run in real-time on a smartphone for table-top environments [86]. In 2010, Wagner et al. [173] presented a real-time approach for the creation and tracking of panoramic maps on mobile phones. This is a useful tool in the context of Handheld AR, where the user often explores the environment by performing rotational movements, remaining approximately in the same spot. In 2011, Pirchheim and Reitmayr presented a real-time camera pose tracking and mapping system which uses the assumption of a planar scene to implement a highly efficient mapping algorithm [132]. In 2013, Gasparini and Bertolino [52] presented a tracking algorithm designed for a mobile device equipped with a stereo camera. In contrast to monocular solutions, no user-interaction is needed for initialization. Reconstruction uses a stereo approach, while tracking runs in real-time on a single camera, but can rely on the stereo setup to generate additional features, when needed. In order to asses the quality of visual tracking, a plane is fitted to the 3D features obtained during initialization. In 2014, Schöps et al. [145] presented a direct (feature-less) approach for tracking and mapping running

on mobile devices. New images are tracked using direct image alignment, while geometry is represented in the form of a semi-dense depth map.

Reconstruction and Tracking of Objects: In 2011, a trend towards the mobile reconstruction of previously unknown 3D objects emerged. This tackles the issue of content creation, which is an integral part in the creation of any AR experience. Besides, the advent of object printing technology further emerged the need for reasonable 3D models of objects. Gruber et al. [55] presented a mobile space-carving approach for the reconstruction of simple objects. Pan et al. [127] presented a novel system employing panoramic images for reconstruction. It allows the generation of a coarse 3D model of the environment within several seconds on mobile phones. In 2013 Prisacariu et al. [136] presented a silhouette-based 3D tracking and reconstruction framework running in real time on a mobile phone. In the same year, Tanskanen [157] showed a dense stereo system for live 3D reconstruction on mobile devices, filling a gap in current cloud-based mobile reconstruction services. Kolev et al. [88] finally developed an efficient and accurate scheme for integrating multiple stereo-based depth hypotheses into a compact and consistent 3D model. Thereby, various criteria based on local geometry orientation, underlying camera setting and photometric evidence are evaluated to judge the reliability of each measurement.

2.2 CV-based Document Inspection

We define CV-based document inspection as the analysis of an instance of a document regarding a given reference model, by evaluation of one or more images for the purpose of getting a decision on its validity. In general, there is a distinction between considering extrinsic (special visual) and intrinsic (mostly textual) features. The first case is relevant for both banknotes and identity documents (see Section 2.2.1), which employ security elements. In this context, an overview of prior art on capturing and inspecting OVDs will be given in Section 2.2.2. The second case targets non-protected documents such as invoices or contracts, which are not considered in this thesis.

2.2.1 Banknotes and ID Documents

Banknotes: There are several works on banknote authentication reported in literature. While most of them use a stationary setup with several different light sources, there are also solutions involving camera phones, which can, of course, only make use of the built-in functionality of the device. Ahmed et al. [1] propose an automated counterfeit detection system for Bangladeshi banknotes, making use of a web-cam and LED light sources (static setup). They use six feature types in order to characterize micro-print, optically variable and iridescent ink (contour analysis), watermarks (PCA), security threads (matching local features) and ultra-violet lines (edge/line detection) present on banknotes. They

accumulate success points for each feature, requiring at least 4 of 6 features for successful authentication.

Machine assisted authentication of Indian banknotes is descried by Roy et al. [142], using a special scanning device (UV, IR, spectrometer) for image acquisition. They mainly use features selected by forensic experts (printing technique, ink properties, thread, art work) and perform classification using both an Artificial Neural Network (ANN) and a Support Vector Machine (SVM) [17]. They compare the scores of the system with those of experts and trained individuals. They report that their method usually gives better decisions than trained individuals, despite being faster.

Bruna et al. [18] build a low cost banknote validation system on custom embedded hardware. They process IR images and tackle the non-uniformity of the light source using a brightness map. Using genuine and counterfeit images of banknotes, they learn the location and appearance of simple patch features including the corresponding thresholds.

A currency reader running on camera phones is presented by Liu [98]. They target especially visually impaired persons. Performing background subtraction and using Boosting on random pair-wise features, the system runs on an off-the-shelf mobile device with a very low false-positive rate. Radványi et al. [137] also perform banknote recognition for the visually impaired using a mobile device (bionic eyeglass). After binarization of the image, tactile marks are detected (morphological operations), the shape of portraits is assessed, and numbers are recognized. At least two consistent votes are required in order to make a decision. By exploiting the topology of documents, the position of missing elements can be estimated including the corners of the banknote. In a user study with visually impaired persons, the system achieved an accuracy of around 96%. Lohweg et al. [103] propose the authentication of security documents and, especially banknotes, with mobile devices using Wavelet-based detection of intaglio printing. First, intaglio line structures are detected and recorded in a categorization map, which form the basis for Wavelet selection. Then, moment-based features are calculated from Wavelet coefficients, but also features based on statistics controlled by the variance (Local Adaptive Cumulative Histogram). Classifier boundaries are calculated using linear discriminant analysis, which they favor over an SVM solution.

Hasanuzzaman et al. [66] perform banknote recognition by computing and matching SURF [11] features on several regions of interest on the document. They require at least two regions to pass and evaluate the approach on 140 dollar bills. They achieve perfect accuracy also with occlusions. Choi et al. [31] classify banknotes from Wavelet features on non-overlapping blocks of edge images. In an evaluation involving 12 classes (10800 images), 99% of Korean bills could be classified correctly. The impact of degraded banknotes is studied by Khashman et al [84]. They use the Discrete Cosine Transform and biorthogonal Wavelet Transform to simulate the desired effects, feeding averaged subimages into an ANN.

ID Documents: Wu et al. [178] present an automatic recognition system for machine-readable travel documents, using textual, facial and identity matching through a database check. Results are combined using fuzzy set-based fusion. Mandridake et al. [106] propose work towards the fully automatic detection of identity fraud. They stress the analysis of background printing and photo substitution and present an initial evaluation of document similarity using SURF features.

Bessmeltsev et al. [12] propose a high-speed approach for reading MRZ data. They analyze the projection profile of the MRZ region to perform slant correction and optical character recognition (OCR) using template-matching. They evaluate their algorithm on a portable passport reader, with a test database of 30 images, achieving perfect accuracy and processing times of less than one second.

Other work in this context targets the automatic verification of signatures written on documents, for which the interested reader is referred to a review paper by Pal et al. [126].

2.2.2 Optically Variable Devices

Related work on OVDs can be divided into approaches suitable for capturing such elements, hologram reconstruction and validation, but also hologram inspection.

Capturing holograms is largely related to capturing a spatially varying bidirectional reflectance distribution function (SVBRDF). This 6D function characterizes the amount of radiance that is reflected at each surface point according to the viewing and lighting directions. Ren et al. describe a portable solution to SVBRDF measurement of flat surfaces using a mobile device, a BRDF chart and a linear light source [141]. Being based on an approach by Dong et al. [34], they locally reconstruct purely specular components, which allows for arbitrary per-point variation of diffuse and specular parts. Jachnik et al. [77] conduct real-time surface light-field capture from a single handheld camera with fixed exposure, shutter and gain. They require a static planar scene and illumination and split diffuse and specular components, finally estimating an environment map. They rely on a guidance component in the form of a colored hemisphere, which indicates whether a pixel has been seen from a particular viewing direction.

The reconstruction of 3D information from holograms is also treated in literature. This is usually connected to digital holography, where an image sensor is used for recording interference patterns, instead of a photo-platter. Buraga-Lefebvre et al. [19] analyze the diffraction pattern on a hologram (in-line holography) using a Wavelet Transform in order to reconstruct the location of small particles in 3D. Their setup requires a laser source, a movable hologram, a relay lens and a camera. They state that diffraction can be treated as a convolution between the amplitude distribution in the object plane and a family of Wavelet functions. In contrast to previous approaches, no focusing on individual particles is required, improving overall accuracy. Amplitude reconstructions of holograms are shown by Pitkäho to be suitable for gaining a depth image using stereo reconstruction [133].

Pramila et al. [135] segment the watermark from a dual-layer hologram. Recording is

done using a camera and a uniform light source, facing towards a tiltable plane containing the hologram. They note that the result is very sensitive to the angle of the plane. Holographic patterns are identified from a printed page by Janucki et al. [78]. They create a Wavelet approximation of the intensity distribution of the hologram and use a Wiener filter to eliminate the influence of non-uniform background. This setup is also suitable for quality estimation of a holographic device. Automatic inspection systems for holograms can use sets of patterns illuminated with multiple IR LEDs on a hemisphere [92, 129]. Images are captured with a CCD camera at controlled illumination angle, and correlation-based matching is carried out in the frequency domain. They extend the system with a correction of rotation angles and evaluate it with two Korean banknotes. Recently, Soukup et al. [151] proposed an approach for sampling the BRDF of Diffractive Optically Variable Imaging Devices (DOVID) using photometric stereo and light-field-based methods. For this purpose, they use a tailored feature descriptor which is robust against several expected sources of inaccuracy, but still specific enough for the given task. They demonstrate their approach on the practical task of automated discrimination between genuine and counterfeited DOVIDs on banknotes.

Besides verification, there have been efforts to combine holograms with computer graphics [14]. By extending a partial hologram reconstruction with additional content, a dynamic high-quality display can be realized [15].

2.3 Document Detection and Classification

Two key issues treated in this thesis are the detection and classification of documents. Although ID documents and, in particular, banknotes get slightly curved when used, in the following, we consider only roughly planar documents. This is mainly motivated by the demand to deliver interactive framerates on off-the-shelf mobile devices (see Chapter 3). Similarly, due to the lack of depth sensing hardware in current mobile devices, only approaches suitable for a monocular setup are considered.

2.3.1 Detection of Rectangular Regions

For detection, we are interested in getting a minimum bounding rectangle of the document region present in an image for further processing. Consequently, the output are the estimated corner positions of a document. Assuming approximate planarity, a document appears in a captured image as a rectangle, which can be perspectively distorted depending on the camera position and orientation. This type of problem is not limited to banknotes and ID-documents, but it is relevant for arbitrary regions fulfilling the aforementioned property. In the following, we present a survey of relevant approaches for the detection of such regions by means of CV.

There are two different approaches for tackling the problem at hand. Either the document corners or borders can be detected directly, or a transformation can be computed

beforehand, with a subsequent detection of borders parallel to the undistorted image frame. In the first case, suitable evidence must be available within the document region.

The estimated transformation can the used remove the perspective distortion of the region enclosed by the detected borders. Throughout this thesis, we mean by the word *rectification* the process of removing this distortion and creating a *rectified* representation through image warping. This is different than in stereo vision [65], where the term *rectification* denotes the process of projecting two or more images onto a common plane using epipolar geometry, while in our case only a single image is processed in general.

Transformation and Detection: Directional information such as text lines or paragraphs available on documents are subject to perspective distortion, depending on the viewpoint. The virtual extension of cues for the same direction appear to intersect in the so-called vanishing point, which is typically outside of the image frame or at infinity, if there is no distortion. If enough cues for directional information are available in the image, such points can be estimated. Using two orthogonal vanishing directions, the image can be fully rectified. Due to the presence of noise on document images, typically a robust estimation is required [89, 131]. It is typically much harder to obtain vertical cues. Miao and Peng [111] use morphological operations to obtain connected components for fitting text lines. After rectification in the horizontal direction, vertical cues are obtained by analysis of character strokes. Yin et al. [180] perform robust vanishing point detection targeting mobile devices. They cluster line intersections and perform a voting operation by projection analysis on the Gaussian sphere space. Alternatively, the rectification of a planar region can also be achieved by using a SLAM approach [132].

The rectified image or a representation thereof (i.e., edge image), can be used for detection of the actual borders of the document region. Zhu et al. [186] detect rectangular particles in cryo-electron microscopy images using a rectangular Hough transform. They split the task into finding the location of rectangles and estimate their orientation. This approach requires all rectangles in the image to have the same size, which must be known in advance. Jung and Schramm [80] use a windowed Hough transform to determine the centers of rectangles and analyze peaks in the accumulator to determine the corners. Bhaskar et al. [13] extract peaks corresponding to line segments from the image using the Hough transform or the Radon transform and perform filtering according to geometric (accumulator space) and spatial (image space) constraints.

Zhang and Kosecka [182] estimate vanishing directions through gradient binning, connected component analysis and line-fitting. They use a homography-based verification of hypotheses to recover the relative pose without calibration. Shaw and Barnes [147] work directly on vanishing lines without requiring a rectified representation. They detect line segments along these directions and perform a set intersection followed by non-maximum suppression through analysis of the neighborhood around the detected locations.

Yonemoto [181] proposes an interactive image rectification method based on the detection of horizontal and vertical lines within a target object. The approach requires user

input by making a horizontal stroke or specifying an entire region of interest.

Direct Detection: If there is insufficient evidence available for computing vanishing points or a feature-based rectification, the document region must be detected directly from the input image. Zhu and Qingzhi [187] perform rectangle detection by chain code tracing and line-fitting. They can also cope with intersecting rectangles.

A common approach for the direct detection of perspectively distorted rectangles is to use image primitives such as corners or line segments. This is usually based on the generation and subsequent verification of hypotheses. An introduction of the general idea can be found in the work of Lin et al. [97].

Tao et al. [8] obtain line primitives from contour tracing and determine pairs of parallel lines, which are merged into rectangles. Lagunovsky and Ablameyko [93] extract linear primitives from lines, group them and perform quadrangle detection by analysis of the distance of line endpoints towards their hypothetical intersection. Li [96] computes edges based on the analysis of smoothed images among all color channels. A minimum bounding box is estimated for each rectangle candidate using exhaustive search (rotation of the support region), which is verified by considering the support in a local neighborhood.

Besides, there are approaches formulating the detection of rectangles in a probabilistic framework using evidence gained through the detection of line segments [101, 112].

2.3.2 Document Classification

Document classification can be seen as a preprocessing step for the subsequent authentication of a document. The goal is to replace or facilitate the selection of the document class by the human operator, giving a list of labels with decreasing relevance.

In general, documents can be represented in terms of their textual content, their visual appearance or their layout. The documents considered within this thesis typically have non-uniform and feature-rich backgrounds, with a varying amount of text and other elements. For ID-documents, a reasonable amount of static and dynamic text is present, including face images and possibly visual security features. However, banknotes have very little text (e.g., serial number) printed on them and often pose less obvious security features. Consequently, it is reasonable to focus on a robust and efficient classification based on the overall appearance instead, possibly fusing additional textual or layout information.

Local Features: Objects captured with mobile phone cameras may differ strongly in appearance, when compared to images obtained in a controlled environment. Consequently, local image features are a reasonable choice for representation, abstracting from custom acquisition conditions. Local image features typically require initial keypoint localization and can be divided into two broad groups. While the first group can be represented as a feature-vector (e.g., SIFT [105], SURF [11]), the second group is computed from pixel differences and stored as a binary string. Binary feature descriptors like BRIEF [20], ORB

[143], BRISK [94] and FREAK [2] can be more efficiently computed and matched. A recent approach called BinBoost [162] finds a low-dimensional, but highly discriminative binary descriptor using supervised learning. The resulting descriptors feature accuracy comparable to floating point descriptors.

Recognition with local features can be realized by feature matching (e.g., based on distance computations). Optionally, the spatial layout can be verified by a suitable model (e.g., robust homography computation using Random Sample Consensus (RANSAC) [46] for the planar case). Chen et al. [29] compute SIFT features of bank forms on a regular grid and use a kd-tree to approximate distance computations for classification. They use 91 classes with only five training samples each, still giving a classification rate of over 99%. In case of a large amount of samples, features can be quantized using a vocabulary of visuals words obtained by clustering all feature descriptors (Bag of Words (BOW)) [149]. Augereau et al. [6] apply the BOW model to visual and textual features with subsequent classification of codeword histograms by SVMs. In order to improve the overall result, they train another SVM with the probabilistic output of the two individual SVMs. They classify 1925 documents from a production chain into one of 12 classes, using just five samples for training. They state that fusion in general gives equal or better results than a single classifier.

However, the vocabulary can become very large, making both storage and retrieval infeasible. By hierarchically clustering of the available features, the BOW model is applicable to problems of larger scale, employing an initial classification [123]. Images are classified using a suitable scoring scheme (e.g., Term Frequency-Inverse Document Frequency (TF-IDF)). Together with subsequent geometric verification of candidates obtained from the hierarchical BOW model, this setup can be used for large-scale mobile recognition of various objects (Mobile Visual Search). Performance can be further enhanced by taking into account the context of local features, but this comes at the cost of increased memory consumption [177]. Initial results obtained by the vocabulary tree may be improved using tree-based re-ranking as an additional step, before performing geometric verification [163]. With a larger number of classes, the dominant factor is the size of the inverted index in the vocabulary tree, which can be compressed [27]. Tsai et al. [166] perform mobile visual search supported by text detection. They identify the title, rectify it and extract the text. This also gives an affine model that can be used to constrain the region of interest and the orientation during the matching stage of geometric verification. Using documents with a large amount of text, the true-positive rate is increased by up to 50%. In later work Tsai, et al. [165] perform mobile visual search on a variant of SIFT computed from HOG information originating from a word patch. They evaluate the approach on a synthetic database of word patches. Although not requiring an OCR-step or dictionary look-up, they achieve similar performance.

Other Approaches: Besides using local features, there are other approaches to classify documents based on their appearance, text information or layout. Gao et al. [49] extract

a dendrogram of MSER regions, roughly corresponding to physical parts of the document. They create a histogram of region pairs encoding the structural relationship and use TF-IDF for classification. In an evaluation containing images of 4,109 invoices from 249 providers, their approach is able to outperform traditional BOW methods. Usilin et al. [167] perform visual appearance-based document classification using modified Haar features [172]. They use only rectangular features and remove the cascade structure, in order to differentiate between five classes of official documents. They are able to achieve a classification rate of 94% using a large amount of training samples.

Van Beusekom et al. [168] perform layout-based classification using pairwise distance measurements between bounding boxes of layout elements. In a subsequent graph-matching step, an optimal assignment is computed and used for the classification of medical journals. While only in 69% of the test cases, the type of the journal was classified correctly, the type of the overall layout was classified correctly in 92% of all cases. Layout-based classification of documents is considered by Cesarini et al. [23]. They use OCR to obtain a set of regions, which are initially represented as a tree-structure and converted into a fixed-size representation for classification.

Besides, there are several approaches targeting the retrieval of documents containing mainly text [40, 99]. Considering the spatial layout of words in a hashing approach, good retrieval performance can be achieved even for large datasets [120, 156]. This setup is even suitable for tracking, thus, enabling the augmentation of a document. Toyama et al. [161] use this approach for image-based document retrieval in an interactive system. They combine it with a wearable eye-tracker and a see-through head-mounted display to determine the region on the document where the user is looking. This allows to automatically retrieve additional information while reading a document and to augment it. They demonstrate the usefulness of the system within a user study, automatically augmenting translations of words onto a document.

Kunze et al. [90] propose an approach for recognizing the type of a document which a person is reading, using an eye tracker. From fixations and saccades, gaze features are computed using a sliding window, and classified with a decision tree. They evaluate the approach with Japanese documents, and succeed in determining the document type (5 classes) in 99% of all cases, provided that the user is known. For an unknown user, the accuracy drops to 74%.

2.4 Text Detection and Recognition

Reading text on arbitrary documents with camera phones is very different compared to processing office documents with a scanning device. ID-documents and banknotes have very diverse backgrounds. In the absence of prior information, the task of text detection and recognition on such documents is actually similar to processing text in natural scenes.

2.4.1 Overview

Standard CV techniques for text detection can be mainly categorized into texture-based and component-based approaches. In texture-based approaches, sliding windows and a subsequent classifier are used for robust detection. However, the computation of text masks for an OCR stage may require additional effort. Component-based approaches tend to compute single characters through segmentation and group them together to form lines or words. Relevant approaches are often rather efficient and provide text masks as a by-product. However, such bottom-up approaches require region filtering operations for improved robustness.

Liu and Sarkar [102] detect candidates for text regions with local adaptive thresholds (binarization). They perform a grouping step considering geometric properties, intensity and shape. Zhu et al. [185] segment and detect text using binarization and subsequent boosting in a cascade for reduction of runtime. Liu et al. [100] use an extended local adaptive thresholding operator that is scale-invariant. Regions are filtered using character stroke features and are then grouped using a graph structure. Color clustering is used by Kasar and Ramakrishnan [82] to produce text candidates. They use twelve different features (geometry, contour, stroke, gradient) in a filtering stage employing an SVM. Milyaev et al. [114] state that reasonable binarization is critical for using off-the-shelf OCR engines. They formulate document binarization in an energy minimization framework to provide a robust basis for end-to-end text recognition.

MSER is used by Merino-Gracia et al. [110] in a system for supporting visually impaired individuals. They employ graph-based grouping on filtered regions for the final result. Donoser et al. [35] use MSER to track and recognize license plate characters. Neumann and Matas [121] extend MSER using topological information and conduct exhaustive search on character sequences, followed by a grouping step and SVM-based validation. They consider careful grouping to be important for getting good results. Gonzalez et al. [3] combine MSER with local adaptive thresholds and also use an SVM-based classifier for the detection of characters.

There are several works which use morphological operations to segment text regions. Fabrizio et al. [43] detect text in street-level images using toggle-mapping and SVM-based validation. Minetto et al. [115] extended this regarding scale-invariance. In addition, a HOG descriptor can be added for improved performance [116].

Epshtein et al. [39] exploit the observation of constant character stroke width using a novel image operator called Stroke-Width Transform (SWT). This approach is based on the evaluation of opposing gradients on the basis of an edge map. They employ several filtering operations to obtain words. Neumann and Matas [122] detect text using extremal regions, which are invariant regarding blurred images, illumination, color, texture and low contrast. Their approach employs a subsequent classification step (Boosting, SVM).

Saio et al. [144] use Wavelet-coefficients for text detection. Mishra et al. [117] first detect characters using HOG features and am SVM in a sliding window. They also use a

lexicon-based prior and combine the available information in an optimization step. Sun et al. [153] evaluate several gradient images and verify the result using a visual saliency operator. Yi and Tian [179] compute regions based on gradients and color information. They propose two different algorithms for grouping, which have a major impact on accuracy. Pan et al. [128] follow a hybrid approach by computing boosted HOG features and binarization with region computation. Opitz et al. [125] perform end-to-end text recognition by computing a ternary coding of local binary patterns [124] over a sliding window and use a Convolutional Neural Network (CNN) [17] for character recognition.

The performance of text detection can be improved by employing visual saliency information. Karthikeyan et al. [81] learn an attention map from several visual saliency features. This gives a binary map that can be used for text detection using SWT, increasing its performance.

An alternative approach for character recognition and detection is to use local features. Iwamura et al. [75] robustly match SIFT features of characters using a local RANSAC algorithm. In this case, the arrangements of local features can be used to roughly determine the corresponding text region in the input image. Kobayashi et al. [87] propose an anytime algorithm, which aims to speed up the aforementioned approach by splitting the input image into cells and prioritizing recognition according to the estimated difficulty of a character and the regions covered by previous characters. They also support video input by tracking characters in order to improve the process. By splitting computations into several threads, a result can be generated once per second on a desktop machine [76]. Matsuda et al. [108] sequentially estimate an affine transformation for each character. This allows the recognition of characters having only three correspondences.

2.4.2 Mobile Systems

Recently mobile devices have become an interesting platform for text detection and extraction[3]. Gomez and Karatzas [54] present a solution for real-time detection and tracking of text suitable for mobile devices. They use MSER for text detection and track groups of regions with multi-oriented text. The system performs text detection and tracking in separate threads and automatically merges results, if new information arrives. They require 1 s for text detection and 40 ms for tracking on an off-the-shelf mobile device.

Wang et al. [176] rely on initial user input for text detection and analyze character stroke width with subsequent refinement through binarization. Fragoso et al. [48] present a mobile system for real-time translation and augmentation of text, which also requires initial user input. They first analyze the gradient distribution near the touchpoint to estimate an initial bounding box. Then, the position of a minimum bounding box is computed using the Hough transform. This can be used to undistort the region. They separate the background, obtaining a mask for OCR. The system finally sends the extracted text to

[3] `http://questvisual.com`

Google Translate[4] and augments the result onto the original region in the image.

2.5 Visualization and User Guidance

Mobile and Handheld AR systems are largely unconstrained setups. Thus, the presentation of information with respect to the real world and guidance for the user deserve special attention. Reviewing the spectrum of security elements present on documents, the inspection of OVDs seems to be the most challenging effort. They require considerable interaction, even when manual inspection is carried out. Since they differ in appearance depending on the viewing direction and the presence of light sources in the environment, it is reasonable to guide the user through the inspection process and to give appropriate feedback.

In literature, there are various works on guiding the user in order to efficiently fulfill a given task, which greatly vary in the size of the workspace, where the interaction takes place. Although for document inspection mainly a subset of the human arm reachable workspace is relevant, we also consider approaches suitable for larger workspaces, given that they could be adapted for the goals of this thesis.

View Alignment: User guidance can be approached by visualization of the view alignment error concerning a given reference pose. Examples are surgical scenarios in telemedicine, where colored augmented coordinates are used for easier navigation of the end effector [30]. Pyramidal frustums can also serve as a means of guidance for navigation. This can be seen as a geometric representation of the camera at the time of capture [150]. This approach is used for real-time visual guidance for accurate alignment of an ultrasound probe by Sun et al. [154]. After tracking artificial skin features for probe localization, visual guidance for 6 DoF alignment is provided via an augmented virtual pyramid. Such a pyramidal representation is also related to the Omnidirectional Funnel [16], which is useful for calling attention. Bae et al. [9] use visual guidance for re-photography. They analyze the camera image to determine, if a sufficiently similar image was captured. Then, three visualizations are presented for alignment. First, a 2D arrow indicates the required direction of movement w.r.t. a top-down camera viewpoint. Second, this information is also indicated concerning a back-front camera viewpoint. Finally, they visualize edges for adjustment and feedback of the current camera orientation. Heger et al. [69] perform user-interactive registration of bone with A-mode ultrasound. The pointer is mechanically tracked, and a 2D-indicator is used to provide visual feedback about the deviation from the surface normal during alignment of the transducer to the local bone surface.

[4] `translate.google.com`

Constrained Navigation: Alternatively, guidance can be achieved by visualization of a constrained navigation space. Shingu et al. [148] create AR visualizations for re-photography tasks. They use a sphere as a pointing indicator along with a half-transparent cone having its apex at the sphere as an indicator of viewing direction. Once the viewpoint is inside the cone, it is not visible anymore. The sphere then changes its color when it is fully visible. This corresponds to a valid recording position. Sukan et al. [152] propose a wider range of look-from and look-at volumes for guiding the user to a constrained set of viewing positions and orientations, not counting roll *(ParaFrustum)*. This can be realized as an in situ visualization or via non-augmented gauges. In the in situ variant, the transparency of volumes is modulated, depending on the distance and orientation of the current pose. In addition, the general representation of the look-at volume is also changed.

2.6 Discussion

Mobile Augmented Reality has a broad track of research and finally found a widely accepted platform in off-the-shelf mobile devices. Real-time tracking using natural features is feasible on mobile phones and can, in principle, be exploited for building advanced document information systems. Document inspection by means of CV focuses on banknotes, but much less on ID-documents or OVDs. Mostly static setups are reported in literature, which target automatic operation. Although they can rely on special hardware and may provide reasonable accuracy, it is not desirable to do without human reasoning [51].

A prerequisite for AR-based inspection is the determination of the document class, which is not trivial to be realized efficiently on the device, given variations in personalization, a large amount of classes, but only a small amount of samples for training. Feature-based approaches give promising results, but, so far, they have not been shown to provide a feasible solution for document classification on mobiles.

Within the broad range of prior art on the detection and extraction of text in natural scenes, there are also mobile solutions. However, the requirements in the context of mobile document inspection are different, in that they call for efficient and accurate end-to-end text recognition, which, in general, cannot rely on a dictionary (e.g., MRZ, serial number).

To the best of our knowledge, there is no mobile approach on the detection or verification of OVDs. Due to their nature, repeatable recordings from several viewing directions must be carried out. Given appropriate lighting conditions, a reasonable strategy for comparing the visual appearance is required. From the perspective of the user, such an approach may require either strict alignment with a given viewing direction or continuous sampling in a constrained space. As literature is limited in this regard, there is room for a custom user-guidance and visualization approach for the inspection of such elements.

3 Document Detection and Classification

Contents

A key ingredient for document inspection is the knowledge about the class of the document in question. Only with this knowledge, suitable reference information can be indexed for subsequent tracking and verification steps. Knowledge about the document class can be gained through an initial classification step. Although the classification can be performed manually by the user, in the context of inspection using Mobile AR, an automatic classification from one or more images of the document in question is desirable for reasons of efficiency and usability.

In a typical Mobile AR pipeline, target recognition and tracking operate on a number of templates known before the start-up of the the application. However, issues arise, when attempting to process personalized documents such as passports or ID-cards in such a setup. Large changes in appearance due to personalization impede robust tracking. Consequently, a standard pipeline for Mobile AR cannot serve as a basic building block for applications targeting document verification.

3.1 Contribution

In contrast to the state-of-the-art, we propose to extract a suitable template directly from the live video stream provided by the built-in camera of mobile devices. So, tracking the document can take place immediately, and the task of determining the actual class of the document is treated as a separate step, which is carried out on the extracted region of the image containing the document (see Figure 3.1).

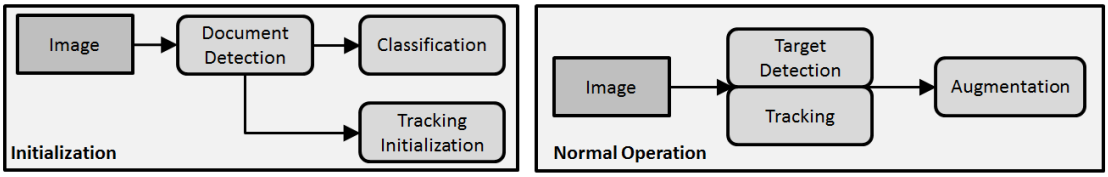

Figure 3.1: Overview of the proposed pipeline for tracking documents with arbitrary personalization.

In the following, we propose efficient solutions for both document detection and classification, which can be computed instantly on off-the-shelf mobile hardware. Besides improving performance regarding tracking and classification, this setup also minimizes undesirable delays and avoids legal issues, which would arise, when performing these steps remotely on the original input images. Even if only a representation needs to be transmitted to the server, this is not desirable for reasons of responsiveness and connectivity. Document detection is handled by a custom solution designed for mobile application, which is not limited to documents, but can be used for an arbitrary rectangular region [63]. For classification, a client-side solution for mobile visual search is introduced, which is modified to allow efficient operation on off-the-shelf mobile devices [64]. In an extensive evaluation, the latter is shown to provide reasonable classification performance on personalized and non-personalized documents, while both computational complexity and storage requirements remain manageable for off-the-shelf mobile devices.

3.2 Document Detection

Printed documents, posters, a deck of cards, the screen of a computer, a window or an image projected onto a wall are all planar objects which are bounded by rectangular borders. Consequently, these are interesting targets for use within mobile AR applications employing NFT to obtain the pose of a target in real-time. Typically, one or more rectangular image targets are delivered with an AR application. Consequently, the targets cannot represent the effects of operating conditions and camera settings on appearance occurring at runtime.

In document inspection, it is required to track a variant of a known target, due to varying personalization. The resulting visual gap degrades tracking performance. Since the entire set of instances is not known, they cannot be deployed with the application, and even if that was the case, the storage requirements would make deployment rather difficult.

This can be solved by instantly creating a tracking target directly on the mobile device by analysis of the video stream. This enables more robust tracking and the use of the extracted target for subsequent image processing tasks, instead of the full frame image.

We propose a method for localization and rectification of a dominant rectangular region within an arbitrary image. This approach can deal with perspective distortion and

Figure 3.2: Top row: Visualization of the region of interest and tracking of the detected and rectified document region within a Mobile AR prototype. Bottom row: Since no assumptions about the contents of the region are made, the proposed approach has several other use cases.

high-frequency structures (see Figure 3.2). We show the success of the approach as well as its applicability to mobile devices in an extensive evaluation on real hardware (see Section 3.2.2). We obtain a significant increase in tracking performance, and are able to substantially improve the recognition rate of an exemplary visual search application by processing a rectified image obtained from the detected rectangular region.

3.2.1 Localization of Rectangular Structures

Rectangle detection using the Hough transform is not robust regarding perspective distortion and harms efficiency, due to the high dimensionality of the accumulator space. We also noticed that vanishing point estimation often fails on typical images showing rectangular tracking targets which do not contain a lot of textual information. Using image primitives resolves these issues. In contrast to previous work [8] [97], we employ adaptive edge detection, filter high frequency noise and deal with a reasonable amount of perspective distortion.

To ensure broad applicability, assumptions on the content of the background image as well as the region to be extracted should be avoided. However, we limit ourselves to planar rectangular regions that are visually separated from the background through their

bounding edges, lying within a region of interest (ROI). This is a trade-off to obtain an algorithm that is computationally feasible on mobile platforms and to improve robustness.

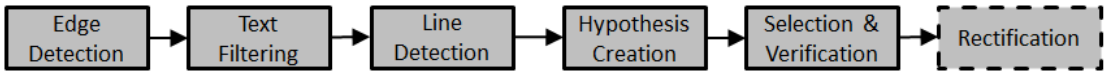

Figure 3.3: Outline of the algorithm: Lines are detected from a filtered edge-map. Then, hypotheses are built from pairs of lines. These are verified using the support on the original edge-map. Finally, the distorted rectangle region is rectified.

3.2.1.1 Algorithm

Initially, we compute an edge map by applying a Canny edge detector [21] with automatic threshold selection (see Figure 3.3 for an illustration of the steps involved). After filtering text-like structures, we perform line detection to obtain all possible lines within the ROI [37]. We group each two lines having a difference in direction of less than 14 degrees together. Then, pairs of line bundles are selected, giving a list of hypotheses for rectangular regions, each consisting of four lines. In order to reduce the amount of initial hypotheses, filtering is applied by intersection of lines and verification that the corners lie within the ROI. This is a critical step that improves robustness as well as efficiency. Finally, the support for each hypothesis is computed on a dilated edge image, which forms the basis for ranking. This is to account for imperfect fitting and a certain amount of curvature. The result is a ranked list of hypotheses, from which the final candidate is taken. This corresponds to the most dominant rectangular region regarding the ROI.

If the region is to be used as a tracking target, we assume it to be rectangular, so that it can be automatically rectified. We compute the dimensions of an undistorted rectangle by averaging the pixel width and height of the corresponding hypothesis. With this information, a homography can be computed for rectification.

3.2.1.2 Text Filtering

High frequency structures can lead to rather strong, but false responses in line detection. Consequently, we perform filtering of the edge image before line detection (see Figure 3.4). This improves robustness as well as runtime of our algorithm by reducing the amount of hypotheses. We noticed that most noise comes from small text-like regions in the input image. Consequently, we compute a rough estimate of such locations by a region-based approach and filter them. We first segment the image using adaptive thresholding [146] and then label it for accessing individual regions including their contours [26]. Each region is then assessed according the criteria aspect ratio, relative height and the amount of pixels with respect to their bounding boxes. We compute each criterion regarding the dimensions of the ROI and apply a suitable threshold for making the final decision on a region. Keeping only the contours of the rejected regions within a mask, we apply a

dilation to account for inaccuracies in segmentation. The edge map of the image is then thinned out according to the obtained text mask, before line detection takes place.

Figure 3.4: Influence of filtering text-like structures on line detection: Result on plain edge map (left). Result on filtered edge map (right).

3.2.1.3 Adaptations for Mobile Phones

As the scenario for image acquisition is largely unconstrained, an automatic selection of parameters is desirable, especially for edge detection. However, the flood fill operation used for hysteresis thresholding in the Canny edge detector was highly inefficient when run on mobile devices. Switching to a stack-based implementation gave a more than 100-fold speed-up compared with a standard implementation. In addition, line detection was a major bottleneck. We achieved a 4-fold speedup by using look-up tables and reducing the resolution of the accumulator space. This allows instant processing on current mobile hardware.

3.2.2 Experimental Evaluation

We first assess the general performance of our algorithm concerning accuracy and runtime. Then, we investigate the performance of the extracted target in NFT. Lastly, we demonstrate that the approach can be beneficial in a visual search scenario. We use the Samsung Galaxy S2 smartphone for all relevant parts of this evaluation (ARM A9-based dual-core CPU up to 1.2 GHz, 1 GB of RAM, 8 MP camera).

3.2.2.1 Accuracy and Runtime

We took 78 images (640 x 480 pixels) of various categories of rectangular items (book covers, business cards, posters) and manually annotated the rectangle corners. For lower resolution images, we downscaled the ground truth locations with the images.

For assessing the quality of localization, we compute correspondences between the extracted and manually annotated corners and the distances between corresponding corners. We define a relative error metric,

$$e = \frac{\Delta d_{ex_{max}}}{d_{ref_{min}}} \tag{3.1}$$

where $\Delta d_{ex_{max}}$ denotes the maximum error in location over four corners, and $d_{ref_{min}}$ denotes the minimum edge length for the annotated quadrangle. According to our results, the relative error decreases w.r.t. the input resolution (see Figure 3.5).

In practice, we found $e = 0.035$ to be a suitable upper threshold indicating successful detection of a target that is usable for all kinds of applications. Nevertheless, the target can be used for AR applications even with a larger relative error.

We evaluated the runtime performance of our algorithm on a mobile device. Figure 3.6 shows the individual processing steps and overall processing time for different input resolutions. Note that the time consumed for the hypothesis filtering step at 320 x 240 pixels is around twice the time required by the two higher resolutions. This can be attributed to merging of high frequency structures, which cannot be covered by our region-based text filtering. Overall, we chose an input resolution of 480 x 360 pixels for the remaining experiments, as it represents a good tradeoff between accuracy and runtime. In specific cases, the lowest resolution still performs adequately, which allows to have near-interactive frame rates on the tested device.

3.2.2.2 Effect on Natural Feature Tracking

We compared tracking performance of a digital representation with that of an extracted target in three conditions: indoor, outdoor and a very low light environment. For this purpose, we implemented a NFT pipeline on mobile phones. We use BRISK [94] as a detector/descriptor in the computation of the initial pose and hand over this information to a patch-based tracker [174]. We process 60 video sequences (640 x 480 pixels) of posters

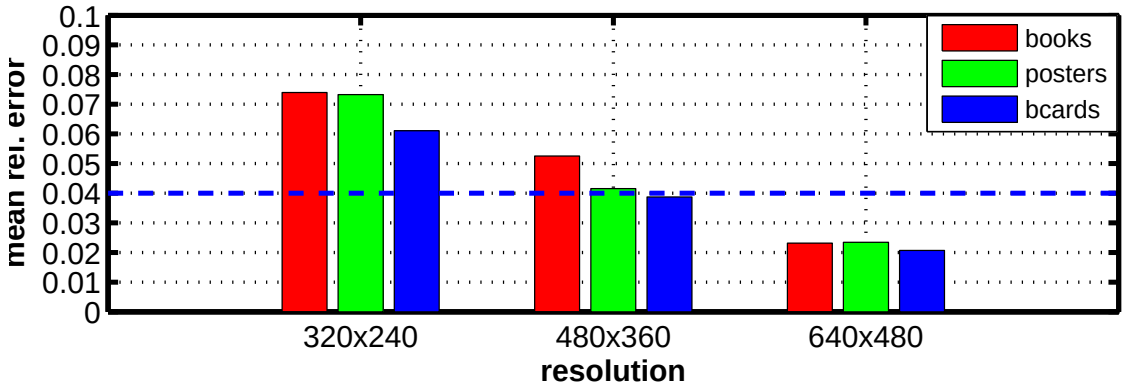

Figure 3.5: Mean relative error per category regarding the input resolution.

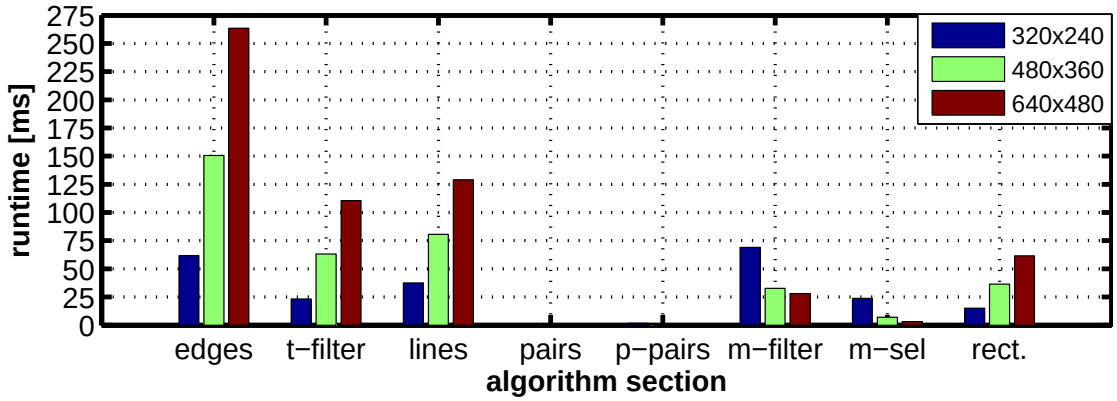

Figure 3.6: Mean runtime for all relevant parts of our algorithm (Samsung Galaxy S2): edges...edge detection, t-filter...text filter, lines...line detection, pairs...parallel pair selection, p-pairs...double pair sel., m-filter...model filtering, m-sel...model selection, rect...rectification

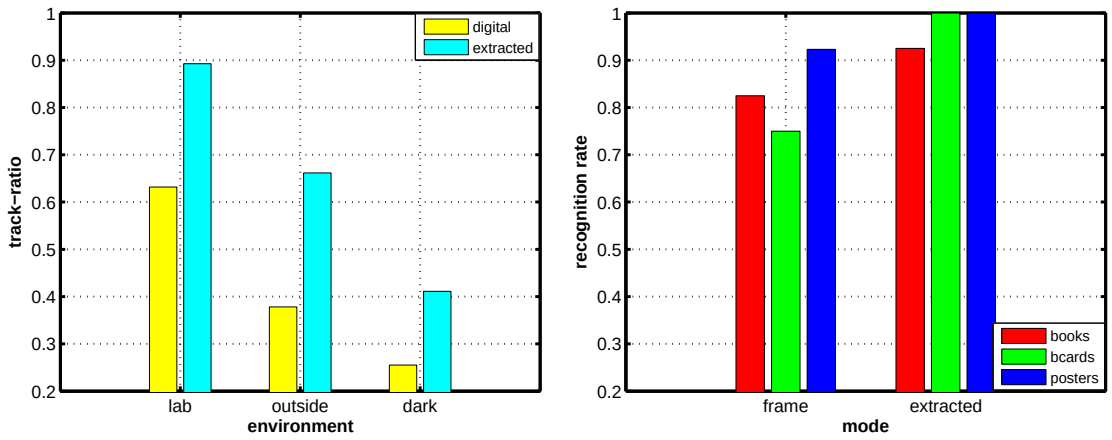

Figure 3.7: Tracking performance w.r.t. environment for digital and extracted targets (left); Recognition rates for visual search using the full image and the extracted region (right)

and books and determine the percentage of successfully tracked frames (track-ratio) with respect to all frames, when using the digital image or a target image extracted from the video (see Figure 3.7). We consider tracking to be successful, if at least th_{kp} percent of all visible keypoints match regarding a given NCC-threshold th_{NCC}. We use $th_{kp} = 10$ percent and $th_{NCC} = 0.68$. In all situations, the extracted target gives a higher ratio of tracked frames. According to our experiments, the expected gain can be more than 25%, depending on the environment of operation.

Similar to tracking, the performance of image recognition can also be improved by processing rectified input images (see Figure 3.7). It must be noted that the business card category is particularly difficult for traditional recognition, because it contains several examples having the same layout but with different personalization. In this case, processing the extracted and rectified image gives a considerable gain in recognition rate.

3.3 Document Classification

The classification step suitable for use in a mobile verification pipeline needs to cover a large number of documents. Still, results must be delivered instantly, despite running locally on the mobile device. This poses additional challenges regarding computational complexity and memory requirements.

In the following, document classification is based on a mobile visual search pipeline, running entirely on the client. Current remote services for visual search have large delays with only little dependence on image resolution (see Section 3.3.5.6). Consequently, it is interesting to investigate visual search from a client-side perspective. We choose a traditional vocabulary tree pipeline for reasons of efficiency, extensibility and popularity. However, we neither transfer the image nor descriptors to a server and perform all processing locally on the mobile device. In particular, the proposed approach contains a number of adaptations to allow instant recognition, when using off-the-shelf mobile devices, improving both accuracy and delay over remote processing.

Since there is no suitable public database available for evaluation, the mobile visual search part is initially evaluated on established databases using full frames. This allows to conduct a comparative evaluation involving a commercial recognition service regarding accuracy and runtime. Then, a series of custom datasets containing rectified documents with different personalization is evaluated with the proposed approach in a separate step.

Disclaimer: It must be noted that the main goal of this part of the thesis was to create a basic building block, which, due to the applied nature of the topic, is required to be able to conduct research in other areas (see Chapters 5 and 6). Thus, we do not aim to fully explore or extend the solution space for mobile document classification, but we provide a *client-side solution*, which fulfills the aforementioned constraints and compares favorably with a commercial server-side solution.

3.3.1 Mobile Visual Search

Visual search is a way of obtaining information about objects in the proximity of the user by taking an image of the object and using the image to index into a database of known objects. The aim of previous work on mobile visual search was mainly to reduce the amount of data that needs to be transferred to a server performing the actual search operation [53] [79]. This applies to the standard pipeline using the vocabulary tree, but also to alternative approaches, which convert the feature-vector to a binary representation [183] or perform hashing [68]. Compressing keypoint locations [164] or using special descriptors further help to reduce transmission time [24]. Still, the initial latency caused by current mobile networks may degrade usability, which is critical in a mobile context.

With advancements in processing power, screen size and connectivity, mobile devices such as smartphones or tablets have become an interesting platform for this kind of service.

It is important to note that, in this case, mobile visual search may replace standard input methods up to a certain degree. This means that information retrieval may take place considerably faster than in a traditional keyboard or touchscreen-driven setup. Today, mobile visual search is available through services like Google Goggles[1] or kooaba[2], dealing with large numbers of categories or classes such as products, logos, printed text, but also places and faces. While the former is available as an application on major mobile platforms, the latter can be queried through a web API, processing a given image.

Prior art closest to the proposed approach is given by Henze et al. [70]. They use heavily optimized local features that are known to sacrifice scale-invariance. The authors only provide a user study on the performance of the system that deals with a rather small number of images. In contrast, we modify selected parts of the pipeline to account for special requirements of mobile setups such as limited processing power and storage capabilities, but also to allow better scaling to a larger number of images. We provide an extensive evaluation of standard datasets with current-off-the-shelf hardware. Thus, we describe a system that is half-way between an online visual search solution and a real-time system. Performing the search locally on the device allows for instant responses, while we are able to limit the memory consumption on current off-the-shelf smartphones for image databases of reasonable size.

3.3.2 Considerations and Approach

On a general level, a major goal of performing mobile visual search on the client is to reduce the large round-trip time of current server-side solutions. Runtime is a very critical factor for mobile applications, and failure to deliver in this area may lead to immediate rejection by the user. Due to constraints in processing power and memory, it is not possible to duplicate a conventional server-side solution onto a mobile device. This also means that the scale of a mobile database will be considerably smaller than a server-side system, as all information needs to be stored on the device itself. The size of applications packages is also critical, as they are typically downloaded by the device over 3G or Wi-Fi networks. Consequently, we need to keep both runtime and storage requirements at a reasonable level, so that the problem remains computationally feasible on current mobile devices. With these considerations in mind, we first implemented a suitable pipeline for visual search and ported it to mobile devices. We then added various modifications so that the pipeline can be used in a realistic scenario employing a large number of image classes or categories, still working in instant time entirely on the mobile device.

3.3.3 Overview

Our pipeline largely follows the standard concept for visual search. We perform keypoint detection and feature extraction on an input image resized to a desired maximum dimen-

[1] `http://www.google.com/mobile/goggles`
[2] `http://www.kooaba.com`

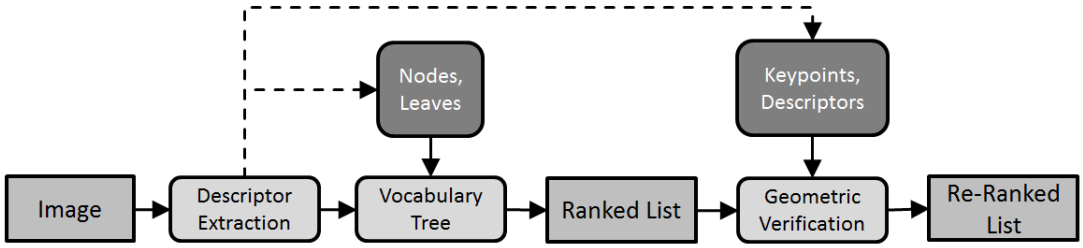

Figure 3.8: Overview of a local visual search pipeline: Image descriptors are extracted and initially classified using a vocabulary tree built from suitable training data. The result is subsequently refined during geometric verification. Dotted lines denote the path of data during the training phase.

sion and hand over this data to a vocabulary tree structure for initial classification. For reasons of efficiency, we perform flat scoring at the leafs. In this step, we pipe each feature descriptor down the tree, accumulate normalized visit counts stored during training for each class and weight them by the corresponding entropy. The final result is reported as the index which corresponds to the maximum of the accumulation vector.

We improve our results in a subsequent verification step with a suitable number of candidates (see Figure 3.8). We serialize both tree data and keypoint/feature descriptor data created during training. While the first is kept in main memory for reasons of speed, the latter is read on demand from flash memory during geometric verification. Since we mainly target planar objects in this work, we employ robust homography estimation using RANSAC to re-rank the list obtained from the vocabulary tree [65].

3.3.4 Modifications for Mobile Application

We made various enhancements to the standard approach for visual search to improve both runtime and memory requirements. Descriptor computation is a critical task in this type of application, as it tends to have a comparatively large runtime. We modified the current implementation of OPEN-SURF [42] by speeding up integral image computation, but also by employing modifications in the final step of descriptor computation. More specifically, we use a grid-size of 3x3, for 36 dimensions in the feature vector. This yields considerable savings in runtime during descriptor computation and geometric verification, but also in terms of storage. Memory consumption is critical, as it influences both installation time and startup time. We reduce requirements in main memory (tree structure), but also in flash memory (keypoints/descriptors) by using half-precision float values throughout the

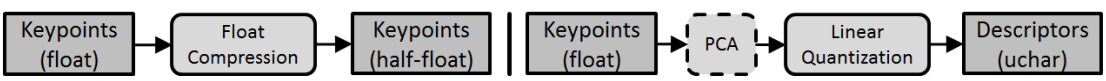

Figure 3.9: Pipeline for compressing local features: Keypoint locations are compressed into half precision values and descriptors are linearly quantized with optional PCA.

pipeline. In particular, this affects keypoint data and vocabulary tree data (see Figure 3.9). In addition, we compress descriptor data by linear quantization into a single byte per dimension. Optionally, we perform PCA [130] to reduce the number of initial dimensions before linear quantization. We also employ compression of inverted index data by recursive integer coding [118], targeting specifically the burden on main memory caused by a large number of image classes. We decompress all data on-the-fly during program execution, working solely on the mobile CPU. We evaluate this client-side pipeline w.r.t. runtime and memory consumption in detail in the next section.

3.3.5 Evaluation for General Purposes

We first evaluate the local pipeline w.r.t. recognition performance, runtime and memory requirements directly on a Samsung Galaxy S3 mobile phone using established databases. This is an off-the-shelf smartphone with an ARM-Cortex A9 CPU (up to 1.4 GHz) and 1 GB of main memory running Android. Information about the performance on this device allows to estimate behavior on most smartphones or tablets currently in use. We evaluate recognition performance using the commercial recognition service kooaba. This allows to compare the behavior of our pipeline to a state-of-the-art solution for image retrieval.

3.3.5.1 Metrics and Datasets

In general, we report recognition rate (relative amount of candidates classified correctly), runtime (descriptor computation, vocabulary tree, geometric verification) and the size of serialized data for the vocabulary tree and keypoints/descriptors. If not noted otherwise, runtime is given in milliseconds (ms) and memory usage is reported in megabytes (MB). Based on informal experiences with acceptable recognition latency, we set the upper runtime limit of a local pipeline at approximately 500 ms on current off-the-shelf devices.

We use several datasets in our evaluation (see Table 3.1). The posters dataset was created mainly to be able to evaluate behavior with various image transformations and serves for initial testing. The Missouri [177] and in particular the Stanford [25] dataset represent typical objects and operating conditions encountered in mobile visual search. Especially the latter is interesting in our context, as it contains more than 1000 classes. Finally, the UK-Bench[3] dataset is included here to be able to evaluate the behavior of the pipeline with a larger number of image classes. This dataset is not very representative for mobile visual search, however, as it also contains different views of non-planar objects, sometimes captured on very textured background. Since there is no test set given, it requires computation of a different metric for evaluation (uky-score).

Although the scale of these experiments is relatively small compared to server-side systems from literature, it seems to be a common practice to create larger datasets by insertion of an arbitrary amount of distractor images. In contrast to our evaluation methodology,

[3] `www.vis.uky.edu/~stewe/ukbench`

Name	Categories	Images	Light	Clutter	Distortion
Posters	11	11	x		x
Missouri Mobile	5	400	x	x	x
Stanford MVS	8	1193	x	x	x
UK-Bench	2550	10200		x	x

Table 3.1: Most datasets used in our evaluation represent typical operating conditions for mobile visual search. The UK-Bench dataset allows evaluations of larger scale.

R train/test	D	P	F [ms]	T [ms]	V [ms]	Sum [ms]
320/320	SIFT	0.8170	1269	21	180	1470
320/320	SURF	**0.8568**	604	9	58	671
320/320	OSURF	0.7829	208	8	59	275
320/320	OSURF36	0.7784	**126**	**4**	**43**	**173**
320/480	OSURF36	0.8409	208	6	34	248

Table 3.2: Local pipeline: Performance and runtime of various local features on the posters dataset: R...resolution, D...type of feature, P...recognition performance, F...feature computation, T...vocabulary tree, V...geometric verification

a comparison is much more difficult in these cases.

3.3.5.2 Evaluation of the Local Pipeline

We first evaluate the system to determine suitable parameters for feature descriptors and geometric verification. Then, we determine the influence of compression on recognition rate, runtime and memory consumption. In a final step, we evaluate our pipeline with a considerably larger number of classes. This allows to come up with a clear statement on performance and practical usability on current mobile hardware.

3.3.5.3 Descriptors and Geometric Verification

We evaluated the influence of the number of candidates used in geometric verifcation on recognition rate and runtime for the posters dataset (see Figure 3.11). We evaluate various feature descriptors for use in our pipeline with the posters dataset (compression switched off). In order to facilitate the comparison of results, we also evaluate SIFT and SURF. We use a maximum extension for the input image of 320 pixels and limit the maximum number of keypoints to 256. Geometric verification is enabled, but configured to just use one candidate. From Table 3.2 and Figure 3.10, it is evident that we obtain reasonable recognition performance with the evaluated feature types. However, runtime of certain setups such as SIFT or SURF is prohibitive for current mobile devices considering our runtime budget of approximately 500 ms. Our modified OPEN-SURF descriptor with

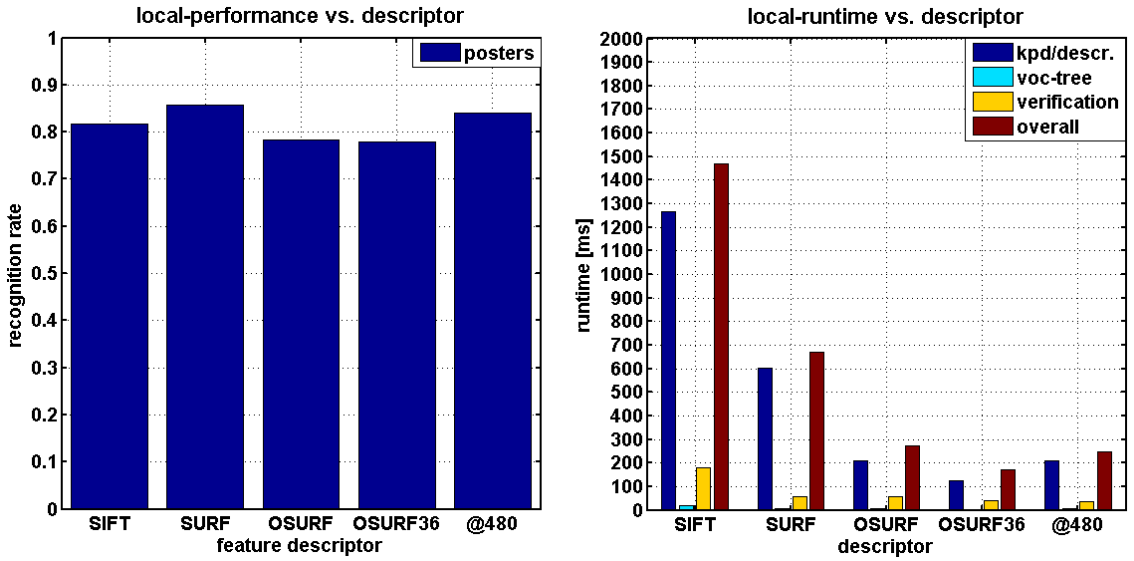

Figure 3.10: Local pipeline: Performance and runtime of various feature descriptors on the posters dataset: Our modified SURF descriptor provides reasonable performance but takes up less runtime compared to the unmodified variants.

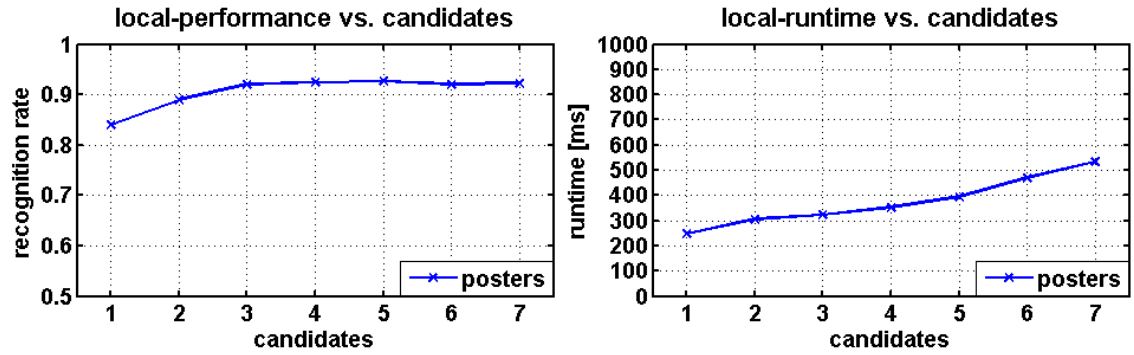

Figure 3.11: Local pipeline: Effect of geometric verification on the posters dataset: Performance saturates around three candidates

just 36 dimensions takes only a fraction of runtime compared to SURF. However, the recognition rate is around 10% lower. As runtime is comparatively low, we can also process images of higher-resolution (e.g., 480 pixels). In this case, we can roughly match the recognition rate of SURF. Still, runtime is less than 50% compared to SURF. In particular, runtime for geometric verification is shorter, which is also due to the reduced size of the descriptor. So, it is possible to use more candidates for a given runtime budget. We see that runtime scales approximately linearly in the number of candidates. Similarly, recognition rate improves with an increasing number of candidates. Although performance seems to saturate, starting with three candidates for the posters dataset, we choose to use six candidates for our modified OPEN-SURF descriptor, as runtime is still around 500

N	D	P	F [ms]	T [ms]	V [ms]	T [MB]	F [MB]
Post.	SURF	0.8568	604	9	**58**	0.98	0.79
Post.	OSURF36	0.9204	212	**6**	135	0.49	0.41
Post.	OSURF36C	**0.9329**	**211**	16	136	**0.15**	**0.12**
Miss.	SURF	0.6751	742	15	**68**	33.2	27.5
Miss.	OSURF36	0.8623	248	**9**	180	16.9	14.6
Miss.	OSURF36C	**0.8759**	**225**	26	151	**5.14**	**4.61**
Stanf.	SURF	0.6550	640	14	**58**	84.1	72.2
Stanf.	OSURF36	0.6940	216	**9**	134	36.4	33.5
Stanf.	OSURF36C	**0.7000**	**216**	26	134	**11.1**	**10.5**

Table 3.3: Local pipeline: Effect of compressing keypoints and descriptors: N...name of dataset, D...type of feature, F...feature computation, P...recognition performance, T...tree, V...geometric verification

ms. Based on our runtime budget, we can only compare the performance of our modified descriptor to SURF using a single candidate for verification. In Table 3.3 and Figure 3.12, we present the results of this setup with the Missouri and Stanford datasets. Compared with the baseline, our modified OPEN-SURF descriptor offers 5%-20% better performance in this setup, but takes up only half of the runtime of SURF. Still, memory requirements for tree and descriptors are comparatively high, especially for the Stanford dataset.

3.3.5.4 Compression

In order to tackle increasing memory requirements, we compress both descriptors and the tree structure. The effect of these measures can be seen in Table 3.3 and Figure 3.12. Our compression efforts significantly reduce memory requirements, while the effect on runtime is negligible. By employing the proposed modifications, up to 85% of storage space can be saved over standard SURF. Interestingly, there is a small increase in recognition performance when compression is enabled. This may be due to a reduction in noise caused by our quantization scheme.

3.3.5.5 Scalability

In this experiment, we determine large-scale performance on the UK-Bench dataset. We perform this test on a Samsung Galaxy S3 smartphone and enable compression of keypoints and descriptors, but also the inverted index stored in the vocabulary tree. From Figure 3.12, it is evident that our pipeline scales well concerning recognition rate, runtime and main memory consumption. It is possible to manage more than 10000 classes with the current pipeline, using less than 110 MB of total storage space. Only a fraction (approximately 50 MB) of data needs to be loaded into main memory. On the one hand, the overall scores obtained in this experiment are lower than those reported in literature, as

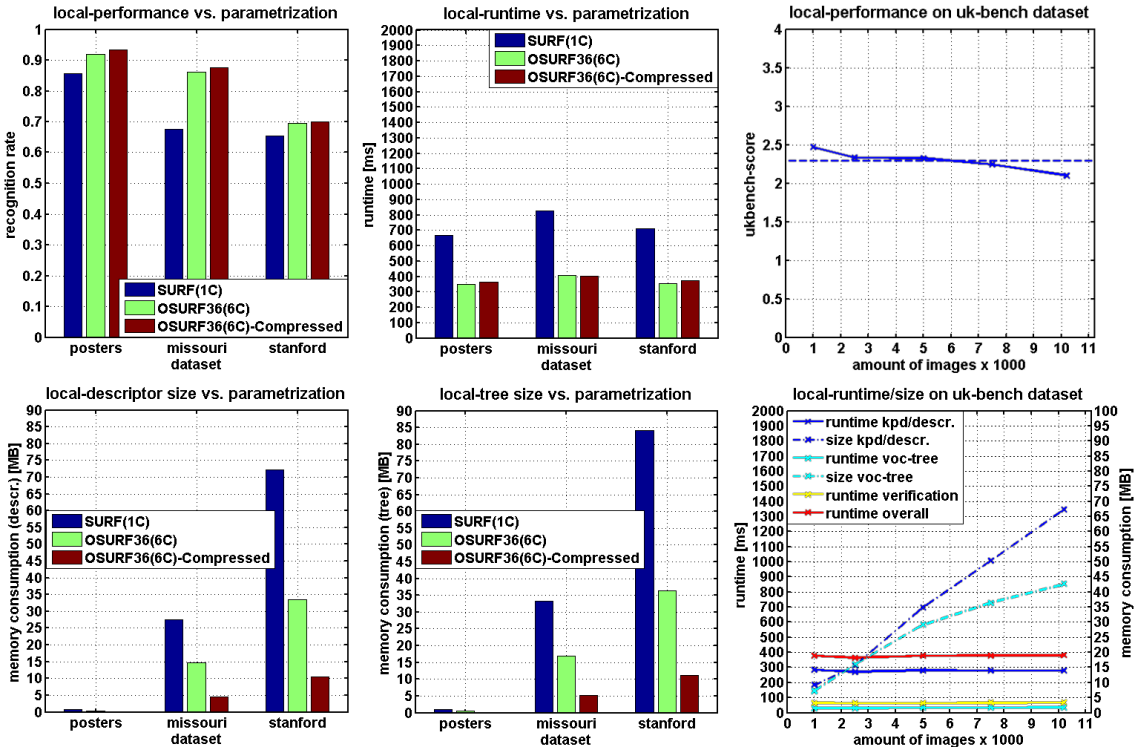

Figure 3.12: Local pipeline: Left and middle column: Detailed evaluation of compression on various datasets: Compared with standard SURF, up to 85% of storage space can be saved at negligible runtime overhead and slightly increased performance. Right column: Testing scalability with the UK-Bench dataset on a Samsung Galaxy S3 mobile phone. Our pipeline scales well concerning recognition rate, runtime and main memory consumption.

our parametrization is targeted towards practical applicability on mobile devices. On the other hand, it does not seem reasonable for this kind of application to train a class for each view of an object. As current mobile devices feature 1-2 GB of main memory and at least 16 GB of flash storage, this purely client-side approach is estimated to be able to handle an amount of images that is around 1-2 magnitudes higher.

3.3.5.6 Comparison with kooaba

For this experiment, we uploaded relevant reference images into a single group and deactivated all images not relevant to the current experiment or dataset. We then performed queries over a Wi-Fi internet connection. This can be considered a very optimistic setup compared to current mobile phone networks.

According to initial tests, the query resolution has little influence on runtime and recognition rate. We scale down query images to a maximum extension of 320 pixels, which is rather common for mobile applications. From Figure 3.13, we see that the posters dataset performs best (approximately 0.7) on kooaba. With the Missouri and Stanford datasets, performance drops by around 10%. Compared to our client-side approach, overall

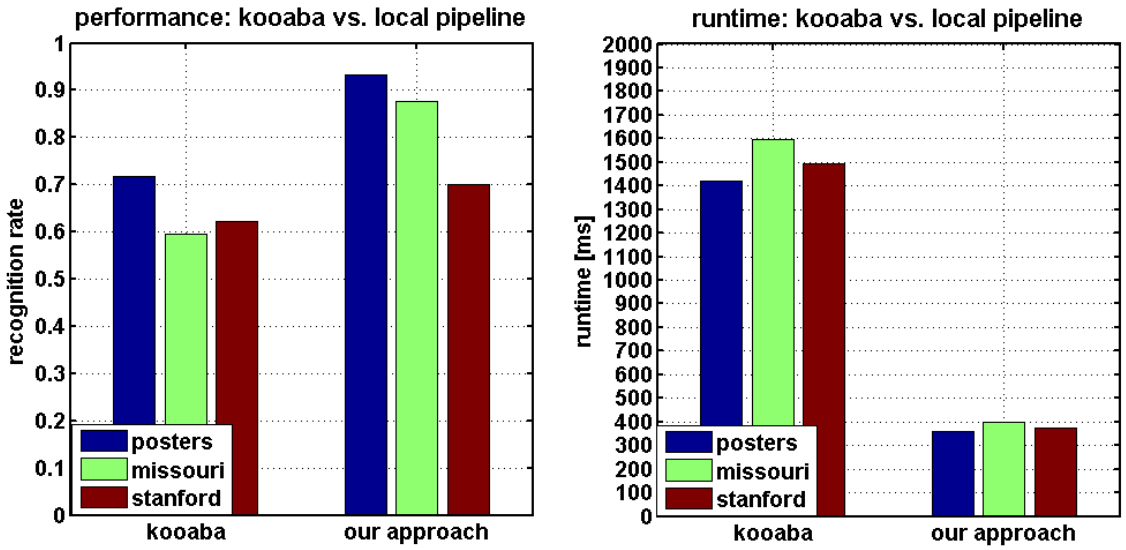

Figure 3.13: Comparison of server-side mobile visual search (kooaba) and client-side mobile visual search: Our client-side solution offers significantly better recogniton and runtime performance.

performance per dataset is significantly lower (5-25%).

However, the client-side approach has lower performance for the print category of the Stanford dataset (see Figure 3.14). As text in general features many keypoints, this drop is likely to be caused by the imposed keypoint limit of our approach. It must be noted that the performance for the landmark category is low for both approaches. This may be caused by the fact that the publicly available training set consists of several images of the same object, each having a separate class. This is not a common application scenario for visual search, however.

For this experiment, runtime of kooaba is around 1500 ms, where the Missouri dataset has a higher runtime than the other two (see Figure 3.13). For our setup, the bottleneck currently seems to lie in the recognition engine itself, rather than connection speed.

All in all, the client-side solution offers significantly better recognition performance on the evaluated datasets compared with a state-of-the-art server-side solution. However, the latter performs better in the print category. A local pipeline giving a result in approximately 500 ms, can, therefore, compete in recognition performance with a server-side solution, which takes three times the runtime.

3.3.5.7 Mobile Prototype Application

We built a mobile prototype for Android smartphones and tablets, demonstrating client-side mobile visual search, but also server-side visual search using kooaba (see Figure 3.15). So, it is possible to compare performance of these approaches side-by-side on current off-the-shelf smartphones. Similar to popular search engines, we give a list of candidates in

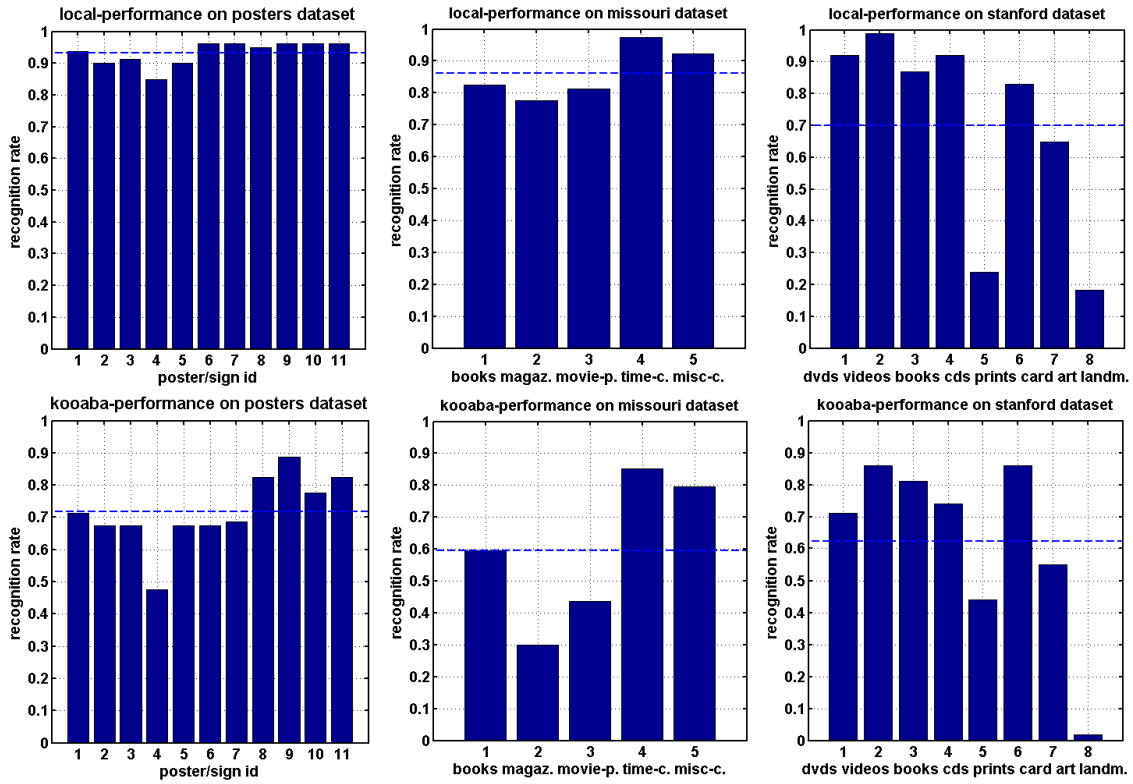

Figure 3.14: Detailed evaluation for the local pipeline and kooaba: Top row: Local pipeline,. Bottom row: kooaba. Our client-side approach delivers significantly improved recognition performance for all datasets and categories except print (keypoint limit).

the form of preview images, which may be activated to trigger a web-based search in order to get additional information.

We also use our client-side module for visual search to extend the amount of realistically usable tracking targets within the Vuforia SDK[4] for mobile Augmented Reality. In this case, we can instantly select a matching tracking dataset without requiring user intervention or costly server-side recognition. We successfully tested this setup with several hundred image targets. Due to limitations in the SDK, we cannot provide a detailed evaluation, however.

3.3.6 Evaluation for Document Classification

The aforementioned implementation of client-side mobile visual search can be used for the classification of documents. In the following, a document class is defined by a background image and generic text attached on top of it. For ID-documents, a unique instance is created using personal data (e.g., image of the owner, textual data, signature).

In the context of document verification, invariance regarding changes in personalization

[4] `https://www.vuforia.com`

Figure 3.15: Mobile prototype: Client-side visual search (top-left). Tracking and augmenting an image target recognized by client-side visual search (top-right). Fast switching of tracking datasets with the local pipeline (bottom-left, 268 ms). Slow recognition using the server-side approach (1984 ms, bottom-right).

and layout is desirable. We carry out an initial evaluation on a small dataset in order to assess the performance regarding such changes with several configurations (see Section 3.3.6.2). Using the most promising configurations, an experiment of larger scale is carried out in order to give a more realistic estimate of performance (see Section 3.3.6.3).

In contrast to common approaches in this context, the chosen pipeline allows to process rectified input images instead of the full input frame. All experiments were carried out on a standard laptop (Intel Core i7 CPU (2 GHz), 8 GB RAM, Windows 8.1).

3.3.6.1 Datasets

There are no public datasets available for an evaluation with documents involving variations in personalization and layout (translation, rotation). Thus, we used synthetically generated data for the major part of the following evaluation. Six datasets of varying scale were used for training or testing. Most of the testsets contain passports, since these are very common documents, and this represents a major use case of the overall approach

ID	Name	Examples	Classes	Synth.	Content
2	did5c1i	5	5	x	passports
4	did5c100i	500	5	x	passports *(personalization)*
5	study11c1i	244	243		passports, ID-cards, banknotes
6	did26c20i	520	26	x	passports *(personalization)*
7	did26c400i	10400	26	x	passports *(personalization, layout)*
8	study11c1im	61	16		passports *taken with mobile phone*

Table 3.4: Datasets used for the evaluation for document classification performance: Terms in brackets denote the type of variations present within the data.

ID-Train	ID-Test	D	DD	C	P	F [ms]	T [ms]	V [ms]	MT [MB]	MF [MB]
2	4	SIFT	128	1	0.998	67	27	34	0.70	0.58
				3	1			104		
		SURF	64	1	0.966	132	18	34	0.46	0.39
				3	1			132		
		OSURF	64	1	1	148	17	19	0.26	0.28
				3	1			81		
			36	1	0.984	112	11	17	0.16	0.14
				3	1			75		

Table 3.5: Initial evaluation of document classification performance: D...type of feature, DD...dimensions, C...candidates for geom. verif., P...recognition performance, F...feature computation, T...vocabulary tree, V...geom. verif., MT/MF...memory consumption for tree/features

(see Table 3.4). Each dataset has an identifier, which is used in subsequent tables to identify the source data used for training and testing. Examples are shown in Figure 3.16 to illustrate the degree of variation in personalization and layout present in the data.

3.3.6.2 Initial Evaluation

The goal of this evaluation is an assessment of performance with variable personalization and changes in layout, using several different types of descriptors. For this purpose, a small training database with just 5 classes (ID: 2) is used. We limited the amount of features for training to a maximum of 1024 and for classification to a maximum of 512. Similar to the results of the general purpose evaluation, the maximum extension of the input image was fixed to 480 pixels. According to Table 3.5, all tested configurations are able to deliver reasonable performance, which can be further improved by using a larger number of candidates in geometric verification.

Figure 3.16: Exemplary images from document evaluation data: Top row: Different personalization. Middle row: Layout variations. Bottom row: Images obtained with a mobile phone.

3.3.6.3 Extended Evaluation

The goal of this part of the evaluation is to assess the behavior of the approach with an increasing number of classes for training and test.

First the amount of classes used for training is increased to a total of 243 (ID: 5), and testing is performed with the same dataset which was used in the initial evaluation. The results are similar compared to the initial test (see Table 3.6). However, the modified SURF descriptor with 36 dimensions is now able to deliver a perfect result and takes only

ID-Train	ID-Test	D	DD	C	P	F [ms]	T [ms]	V [ms]	MT [MB]	MF [MB]
5	4	OSURF	64	1	0.998	156	27	21	11.4	10.2
				3	1			73		
			36	1	1	104	15	16	7.4	6.75
				3	1			59		
	6		64	1	0.9980	157	27	19	11.4	10.2
				3	0.9980			83		
			36	1	0.9940	113	16	16	7.4	6.75
				3	1			74		
	7		64	1	0.9666	176	26	23	11.4	10.2
			36	1	0.9422	116	15	21	7.4	6.75

Table 3.6: Extended evaluation of document classification performance: D...type of feature, DD...dimensions, C...candidates for geom. verif., P...recognition performance, F...feature computation, T...vocabulary tree, V...geom. verif., MT/MF...memory consumption for tree/features

7.4 MB of memory for the tree structure and 6.75 MB of memory for verification data.

When evaluating the impact of different personalization (ID: 6), recognition rates drop slightly to 0.998 for OSURF or 0.994 for the modified descriptor. When using three candidates, the perfect result of the previous evaluation is preserved.

Then, all available synthetic data is used for testing, which are 10400 samples (ID: 7). Besides different personalization, also layout variations (see Figure 3.16) are contained within the data.

All the previous tests gave usable results given a reasonable parametrization, but they were all using synthetic data. For a more realistic assessment of performance, also images of real documents captured with mobile phones must be considered. We used all the available samples and captured them using the Galaxy Nexus smartphone. Then, the document detection and rectification approach described in Section 3.2 was carried out in order to get a rectified input image (ID: 8). According to Table 3.7, geometric verification has a big impact on recognition performance in this case. Considering only a single candidate, the best result is obtained by the unmodified OSURF descriptor (0.7704). When using three candidates, the recognition rate is already 0.9344. This is about 10% more than the modified descriptor. Considering five candidates from the list obtained with the modified descriptor, the performance is slightly above 0.9. However, this saves several megabytes of main memory. The progress of the recognition rate with one, three or five candidates is plotted in Figure 3.17. Since the recognition rate increases when considering more images from the result list, it may still be feasible to use a single candidate for verification. However, this would cause additional effort for the user, who needs to perform the final selection from the obtained list of results.

ID-Train	ID-Test	D	DD	C	P	F [ms]	T [ms]	V [ms]	MT [MB]	MF [MB]
5	8	OSURF	64	1	0.7704	146	25	29	11.4	10.2
				3	0.9344			71		
				5	0.9508			111		
			36	1	0.6721	105	15	21	7.4	6.75
				3	0.8360			60		
				5	0.9016			97		

Table 3.7: Evaluation of mobile visual search on rectified images (Samsung Galaxy Nexus smartphone): D...type of feature, DD...dimensions, C...candidates for geom. verif., P...recognition performance, F...feature computation, T...vocabulary tree, V...geom. verif., MT/MF...memory consumption for tree/features

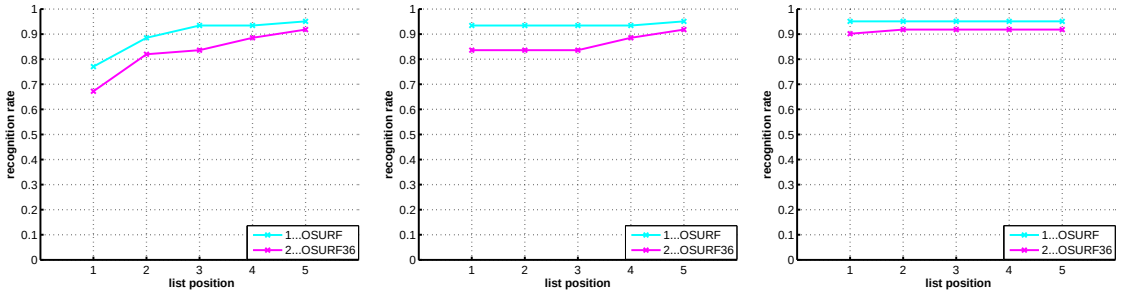

Figure 3.17: Evaluation results using rectified document images (Samsung Galaxy Nexus smartphone) considering one (left), three (middle) or five (right) candidates during geometric verification. Although the recognition rate increases with more candidates, using a single candidate can still be feasible. In this case, the final selection must be done by the user.

3.4 Conclusion

We approached the requirement of tracking and classifying documents with arbitrary personalization by splitting up the traditional pipeline for mobile AR. By first carrying out a detection step on the input frame, the borders of the document are localized. After rectification, we classify it and use the extracted image as a tracking target.

Detection: The document detection approach is realized by searching for a dominant rectangular region within the input image. No assumptions are made about the content of the region, except that it is bounded by borders which correspond to edges in the image. Through suitable modification, this algorithm runs in instant time on off-the-shelf smartphones, which we demonstrated throughout an extensive evaluation. Building on reasonable accuracy in detection, using the rectified region improved the success rate of tracking by more than 25% and the recognition rate by around 10%, with even larger gains for certain document categories.

We observed that grid-like structures can lead to false positives. This could be resolved by making further assumptions about the content of the document (e.g., vanishing point estimation from text lines) or by processing several frames and fusing the result. When enough additional cues can be gained from the input image, a full rectification can be carried out, preceding the actual detection of the document borders. An available text mask can directly be used for filtering the edge map before line detection.

Alternatively, trying to track several hypotheses could help to find a valid detection result. A perpendicular edge search could obtain a better estimate with curved contours. While possibly making the process more accurate or successful, the effect on runtime needs to be investigated.

Classification: For the classification task, a client-side solution for mobile visual search is employed. Starting from a traditional visual search pipeline, several optimizations were made in order to decrease runtime and reduce storage space. We evaluated this approach first with full-frame images instead of rectified ones to assess its general performance. Compared with a standard approach, 85% of storage space can be saved, delivering considerably increased recognition performance at only a fraction of the runtime of a commercial solution. Besides classification of single-images, this approach can also be used as a pre-selector for tracking databases in a commercial AR solution. The scale of the solution can be increased without requiring user intervention or costly server-side recognition.

We evaluated the proposed classification approach with undistorted document images. Synthetic data including variations in personalization and layout was evaluated in several configurations. With a suitable number of candidates used in geometric verification, classification rates exceeding 90% can be achieved. A subsequent evaluation with images taken with an off-the-shelf mobile phone confirmed the eligibility of the proposed approach for practical application.

For future work, descriptor computation should be accelerated further, possibly by using the GPU for part of the processing. This would certainly lead to an even more responsive system, but might also improve the recognition rate by relaxing the current limitation on the number of keypoints/feature descriptors. As their size again poses a problem for huge numbers of classes, they should be further compressed (e.g., variable-rate quantization).

4

Detection and Recognition of Machine-Readable Zones

Contents

Identity documents originally had to be read manually for querying additional information about an individual from a database. In 1968, efforts were initiated to establish a possibility for an automatic reading procedure, with the goal to speed up identity checks and to avoid human error in reading textual identification data [74]. This lead to the introduction of a machine-readable zone (MRZ) found on documents such as passports, visas and ID-cards during the 1980s (see Figure 4.1). The underlying specification was subsequently revised and the MRZ still plays a major role regarding current ePassports, where a valid MRZ reading is required in order to gain access to the chip inside the passport. This in turn ensures that the passport must be accessible for visual inspection when attempting to access the stored information. The information contained in the MRZ can also be used to classify a document, which is required for a subsequent verification task.

The information contained in the MRZ is an extract of the contents of the Visual Inspection Zone (VIZ) with additional local and global checksums. They serve as a measure for data integrity, which could be infringed due to reading errors or deliberate modification of the contents. There are three different types of MRZ, usually placed on the identity page of machine-readable travel documents or the back side of ID-cards. They consist of a paragraph with two or three parallel lines of black OCR-B text (fixed width and size) with fixed inter-line distance (see Figure 4.3). The characters are additionally required to respond in the near-infrared spectrum and may be combined with other security features, provided that there is no adverse effect on reading quality [74]. While most of the contents

Figure 4.1: Exemplary documents with machine-readable zones (MRZ). Left: Identity card. Right: Passport. The MRZ region is depicted by a red rectangle overlaid onto the original image.

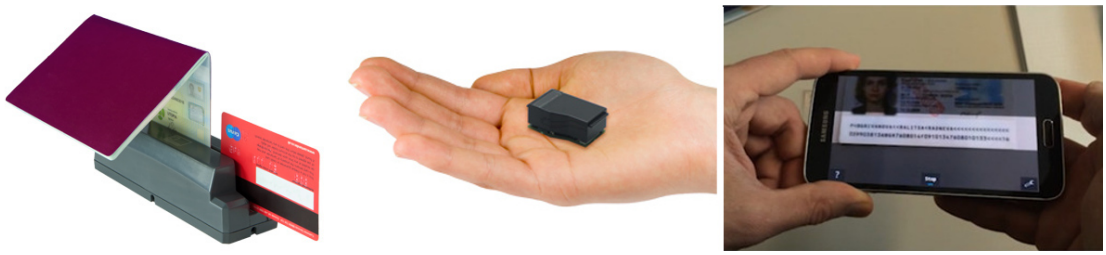

Figure 4.2: Reading solutions for MRZs. Left: Stationary reader. Middle: Snap-on device for mobile phones. Right: Mobile application using the build-in camera of a smartphone requiring an initial alignment operation.

of the MRZ are specified, especially the TD-1 format contains unoccupied space for placing additional data.

Reading MRZ data usually requires dedicated machinery, be it stationary or mobile. In the context of mobile application, there is also additional hardware, which can be attached to standard mobile phones[1] [2] (see Figure 4.2).

Besides, there are mobile applications, which claim to support robust reading of MRZ data from the built-in camera of the device (Smart 3D OCR MRZ[3], ABBY on Device OCR[4], Keesing AuthentiScan[5] or Jumio FastFill/Netverify[6]). All available approaches using the built-in camera have in common that the MRZ must be aligned with the image capture device, before the actual reading operation can take place (see Figure 4.2). This requirement prolongs reading time and runs against the original intention of machine-readable travel documents.

Consequently, there is a need for a more flexible solution of reading MRZ data on off-the-shelf mobile devices. Although the MRZ is designed to be read by automatic

[1] http://www.access-is.com
[2] http://www.movion.eu/grabba
[3] http://smartengines.biz
[4] http://www.abbyy-developers.eu
[5] https://www.keesingtechnologies.com
[6] https://www.jumio.com

machinery, solving the task in a general setting, as proposed in this work, is far from trivial, as is the character recognition. As there is no prior knowledge about the presence of a MRZ, the algorithm has to identify the area of interest automatically in real-time, despite motion blur and all other adversities emerging in real-world mobile phone image acquisition. The subsequent character recognition algorithm is challenged by the need for perfect recognition performance, to make the overall system competitive.

4.1 Contribution

We propose a solution for detecting and recognizing machine-readable zones on arbitrary documents using off-the-shelf mobile devices without additional hardware [60]. In contrast to current mobile applications that use the built-in camera, our approach does not require strict alignment with an orthogonal viewing direction at a fixed distance. By performing an initial detection step and assessment of the capture conditions, visual feedback about the status of the operation can be continuously provided to the user. We show that our algorithms allow real-time detection and instant reading of the relevant information from arbitrary documents (see Figure 4.9). This provides the basis for an efficient and cost-effective way to read and check the validity of MRZ data, which realistically cannot be done manually during document inspection. Together with the communication capabilities of current mobile devices, off-the-shelf smartphones can be turned into devices for querying additional information about an individual from appropriate databases.

Since there is no publicly available database for developing and evaluating MRZ reading algorithms, we also contribute a large database of synthetic MRZ data, covering a broad range of diverse acquisition settings, backgrounds and view points. The database is used to evaluate our approach, giving a baseline for future developments in MRZ reading.

4.2 Algorithm

We identified a set of properties for text on documents - in particular for the MRZ - which are useful for detection and reading. Text regions on documents are generally much smaller than text-like distortions in the background. A local region containing text normally consists of a single color with limited variation, and the stroke width of each character is roughly constant. All character boundaries are closed, and connecting lines on the contour are smooth. These boundaries correspond largely with edges detected in the input image. Single characters within text regions generally have very similar properties and are connected along an oriented line. In most cases, a minimum number of characters per text region can be assumed.

The approach we suggest for mobile MRZ reading works in four steps. First, the location of candidate text must be determined in the image. From this information, the MRZ is detected by considering the spatial layout between candidate groups. Then, a

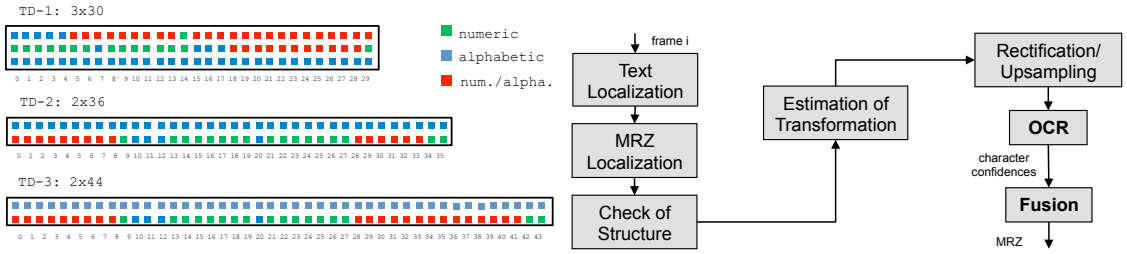

Figure 4.3: Left: Structure of machine-readable zones. There are three different types, which contain two or three lines of text. This corresponds to 90, 72 or 88 individual characters. Right: Outline of our algorithm for mobile MRZ reading. The MRZ structure is detected from text groups. Then, individual characters are rectified using an estimated transformation and fed into a custom OCR stage. Several frames are fused together for better performance.

Figure 4.4: Steps in our algorithm. Top row: Input-image, segmentation result, filtered connected components. Bottom row: Delaunay triangulation, filtered pairs, final MRZ detection result.

local transformation for each character is estimated, which can be used for rectification, followed by the recognition of characters, giving a confidence value w.r.t. each character of the relevant subset of the OCR-B font. Finally, information from several input frames is fused in order to improve the result (see Figure 4.3). We will now discuss these steps in more detail.

4.2.1 Text Detection

We employ Toggle Mapping [44] and linear-time region labeling [26] as basic building blocks for initial generation of connected components (see Figure 4.5). Initial filtering is done based on region geometry and boundary properties (area, extension, aspect ratio, fill ratio, compactness). We also experimented with edge contrast and stroke width, but

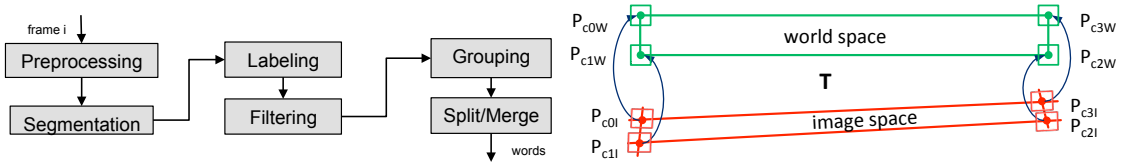

Figure 4.5: Left: Outline of the text detection approach used in our framework. Connected components are obtained from an initial segmentation step, labeled and filtered. Then, they are pair-wise grouped and split into words, providing the basis for MRZ detection. Right: Rectification of Characters: First, a global transformation T is estimated using intersections points of fitted lines in image space and the corresponding world coordinates. Then, a local transformation can be estimated per character, which is then used for patch warping.

these did not improve results significantly at that stage.

Similar regions are grouped together based on region-properties and spatial coherence of characters. For reasons of efficiency, a Delaunay triangulation is used for getting an initial pair-wise grouping. Pair-wise connections in the graph are then filtered using various relative criteria (height, distance, position-offset, area, angle, grey-value, stroke-width) followed by generation of strongly connected components [158]. This gives a series of ordered groups, ideally representing single text words, but, depending on parametrization and document structure, several words can be contained (see Figure 4.4). Therefore, an additional filtering step is employed.

In a split/merge approach based on group properties (min. number of components, max./min. distances, direction, grey-value, area, stroke-width), final text groups are generated. From the filtered groups, the individual components of the MRZ can be detected by analysis of their geometry. We search for groups fulfilling a minimum length requirement (30 characters). During selection, their horizontal and vertical distances are analyzed, finally giving a number of groups that are considered for processing in the optical character recognition stage.

4.2.2 Rectification

The detected characters can be rectified using MRZ structure information (see Figure 4.3). First, horizontal and vertical lines are fitted onto the detected MRZ components using linear regression on their centroids. These lines are further intersected in order to give improved estimates of the four outermost character centers P_{cI}. Using the known properties of the OCR-B font, corresponding coordinates P_{cW} can be computed in rectified (world) space, which allow to estimate a perspective transformation T. For each character centroid, as obtained from the intersection process, the limits of the patch can be determined in world space using font properties and projected into the input image. Now a local transformation can be estimated for each character, which can be used for rectification (see Figure 4.5). In order to improve the input for the OCR stage, we perform up-sampling of character patches during warping.

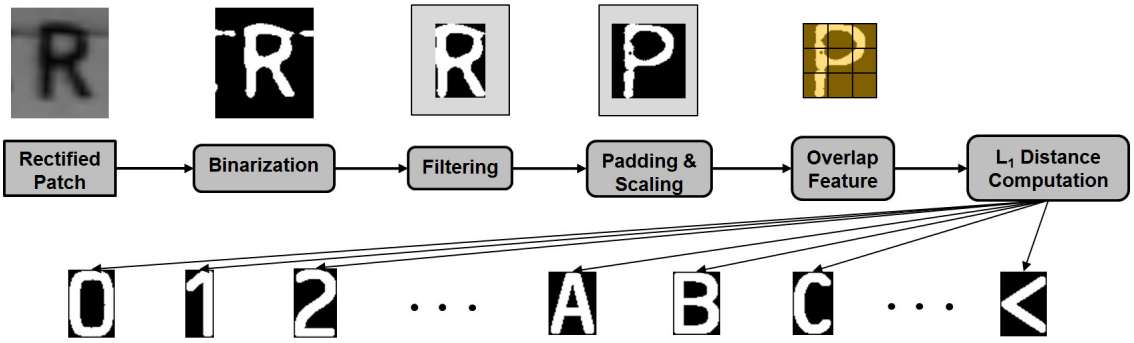

Figure 4.6: Overview of the OCR stage: Rectified patches are binarized, filtered and padded. After scaling, an overlap feature is computed, which gives a feature vector for further classification.

4.2.3 Optical Character Recognition

The OCR stage uses the result of a subsequent binarization step as input data. We use Toggle Mapping for this task, label the obtained binary mask and estimate a minimum bounding box for the character (see Figure 4.6). Through a careful selection of frames, a small number of samples is sufficient for the recognition of single characters.

We employ an overlap-metric for character recognition, which is computed on a regular grid [73]. We compute the local overlap for each cell and store it as a feature-vector. Using the L_1 distance, the similarity concerning a number of reference templates can be computed, which is also treated as a confidence value. We use ARM NEON[7] instructions in the matching stage in order to be able to deal with a higher number of template characters. We generated the initial samples by rendering true-type fonts and added a small number of real samples, which were extracted using the proposed approach.

4.2.4 Frame Fusion

When working with live-video, several frames can be processed on the mobile device for improving robustness. For a subsequent fusion process, correspondences between characters must be established. In the fashion of tracking by detection, the structure of the initial detection result is considered whenever searching for suitable frames.

In each frame i, for every MRZ character j, distances $d_{i,j,k}$ concerning all known references k can be recorded. For each entry, the mean value w.r.t. all frames is computed: $d_{j,k} = mean(d_{i,j,k})$. The final result per character is then computed as the one having the smallest distance: $d_i = min(d_{j,k})$.

[7] http://www.arm.com/products/processors/technologies/neon.php

4.3 Synthetic MRZ Dataset

Due to legal issues, it is not possible to get hold of a large number of identity documents for evaluation. Therefore, a large database for developing and evaluating MRZ reading algorithms is not publicly available.

We collected a set of different ID documents and passports from Google images, using only images marked as free for modification and distribution, sorted them according to their MRZ type and systematically removed the MRZ through impainting. We then use these document templates with different backgrounds and render both the document and a randomly generated MRZ string of the corresponding type. The MRZ string is generated by leveraging a public database of common names[8], using different nationality codes and adding a random time stamp as the birth date, the date of issue and the date of expiry. Finally, the MRZ is completed with the corresponding checksums [74]. Through this generic approach, we can create any number of example documents, single images and also entire frame sequences. In total, over 11.000 different MRZs were generated, resulting in more than 90.000 individual images. In contrast to a preceding evaluation involving only images of 640 x 480 pixels [60], we created a new database containing images with higher resolution (1440 x 1080 pixels) and re-evaluated an updated version of the algorithm with unified character warping and extended reference data (111 samples). Images were scaled down to the same resolution used in the initial version (640 x 480 pixels), before being fed into the algorithm.

Single Images: To generate realistic views of the documents, typical viewpoints are simulated by transformation and rendering of the current template-MRZ combination. In order to mimic typical user behavior, small local changes in transformation are introduced to create a number of images around a selected global setting. Noise and blur is added to the rendered document to increase realism. These documents are considered for the

[8] https://www.drupal.org/project/namedb

Figure 4.7: Single MRZ documents placed in front of a cluttered background image. Backgrounds with different complexities are used, starting from almost uniform to completely cluttered.

Figure 4.8: Top: Sequences of frames rendered onto a random background, and the corresponding camera trajectory. For better visibility, only every 25^{th} frame is drawn as a frustum. Bottom: Sample frames from two sequences. As the document is rendered into a video, the background changes with each frame.

evaluation of algorithms based on single snapshots. Some sample images are depicted in Figure 4.7. To also allow for ID document *detection* algorithms to work on the proposed dataset, different backgrounds are used to reflect different levels of detection complexity.

Image Sequences: As mobile devices can be used to acquire entire frame sequences dynamically, we also created a set of frame sequences. We recorded several motion patterns of a mobile device over a planar target, storing the calculated pose for each frame [174]. The average length of these sequences is about 100 frames. For each frame, we render the template-MRZ combination using the previously recorded pose onto frames from a video taken at a public train station. Thereby we also allow the evaluation of approaches which are able to detect and track a document and combine the reading results over multiple frames. Sample camera paths and corresponding rendered image sequences are shown in Figure 4.8.

4.4 Evaluation

In the following experiments, we determine the accuracy of MRZ detection, character reading and runtime for all relevant steps of the proposed approach. We evaluate a prototype of the MRZ reader (see Figure 4.9) on various mobile devices running Android with captured images of real documents and synthetic images from the aforementioned database. In contrast to previous work [60], we evaluate a revised implementation of the

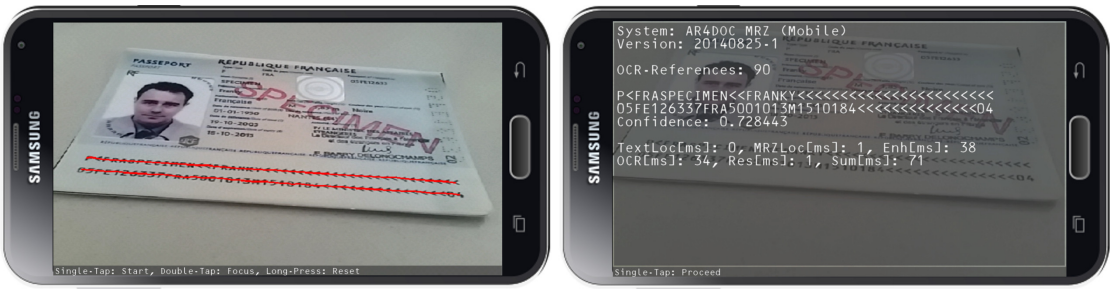

Figure 4.9: Prototype application for reading MRZs without strict requirements regarding the viewpoint (Samsung Galaxy S5). Left: Detection phase. Right: Final reading result.

algorithm on a re-created database featuring images of higher resolution.

4.4.1 Initial Experiments

We conducted a series of initial experiments with real image sequences acquired with the Samsung Galaxy S5 smartphone from twelve different documents. In order to reduce lag within the mobile application, we initially processed VGA images in our pipeline. Then we switched to a HD-like resolution maintaining a 4:3 aspect ratio (1440 x 1080 pixels). These choices are motivated by the available formats of real-time video streams on current mobile devices. In both cases, the images were down-scaled for text detection and transformation estimation, whereas the real input resolution was used in the OCR stage. It must be noted that we considered only MRZ readings from successful detection results.

We see from Figure 4.10 that considerable gains can be made for MRZ readings from VGA images by fusion of results from several frames. While for a single frame just 75.00% of all readings are fully correct, starting with two frames, this number is increased to 91.67% and gives a perfect result with five frames. With images of higher resolution, there is a much smaller difference when fusing reading results over more frames (91.67% for two frames vs a perfect reading result starting with three frames). From these results it is evident that the fusion operation helps to improve reading results for both smaller and larger resolutions. Consequently, is is sufficient to process several images of lower resolution, which translates to a more responsive application.

4.4.2 Reading Accuracy

Reading accuracy is evaluated using single and multiple frames taken from the single-image database. While individual character recognition is barely affected by using more frames, the performance of MRZ detection is noticeably increased (see Figure 4.11). A MRZ detection rate of 88.48% is achieved by using five frames, along with a character reading rate of 99.96%. In terms of successful detection, this is a significant improvement over processing single shots (detection: 78.16%) on low-resolution mobile-images in real-time.

Although the detection and reading of individual characters works reasonably well,

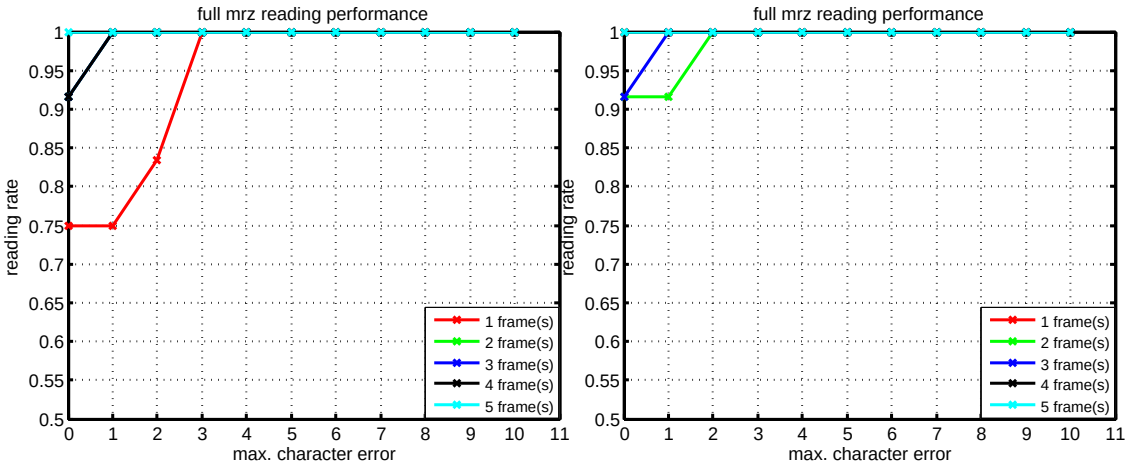

Figure 4.10: Initial Experiments: Full MRZ reading for images of 640 x 480 pixels (left) and images of 1440 x 1080 pixels (right). The fusion operation helps to improve results in both cases. When fusing reading results from five images of lower resolution, a perfect reading rate can be achieved on the evaluated dataset.

getting correct readings for the entire MRZ is still a challenging task, since no dictionary can be used for large parts of the MRZ (see Figure 4.11). However, frame fusion helps to improve the results by up to 4%.

Obviously, MRZ detection performance and character reading are related to the input pose (see Figure 4.13). We can observe that the proposed approach can detect and read MRZ data despite perspective distortion, saving document alignment time for the user. Most gaps seem to be caused by segmentation artifacts, which cause unresolvable ambiguities in the grouping stage. However, the largest gap for the exemplary sequence consists of just two frames, which corresponds to a maximum waiting time of less than 0.1 s for getting processable data, or less than 0.5 s when fusing five frames (assuming a framerate of 30 FPS).

4.4.3 Algorithm Runtime

Runtime is dominated by the OCR part of the algorithm, the rectification, segmentation and feature computation (see Figure 4.12), while the initial text detection and subsequent fusion operations take up only a fraction of the overall runtime.

In total, reading a single MRZ takes around 270 ms on the NVIDIA Shield tablet. On the Samsung Galaxy S5 smartphone, it takes 361 ms per frame, whereas on our development machine (MBP i7, 2 GHZ), the overall runtime per frame is around 77 ms. It must be noted that an optimized built for the Shield tablet was used in obtaining the aforementioned measurements.

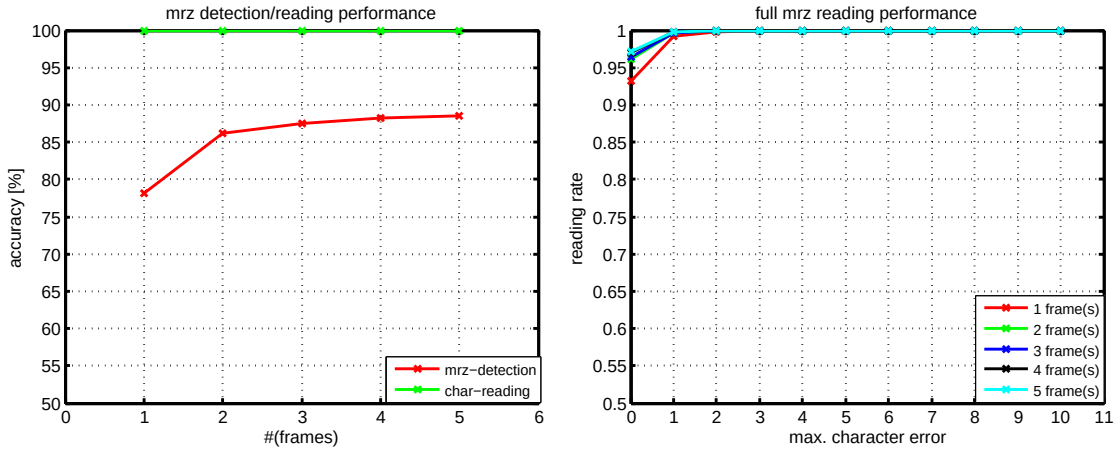

Figure 4.11: Left: MRZ detection and character reading accuracy (single-image database): While individual character recognition is barely affected by using more frames, the performance of MRZ detection is noticeably increased. Right: Full MRZ reading accuracy (single-image database): Despite reasonable character recognition rates, reading entire MRZs is still difficult, since no dictionary can be used for most parts. However, fusion of reading results from several frames improves reading rates by up to 4%.

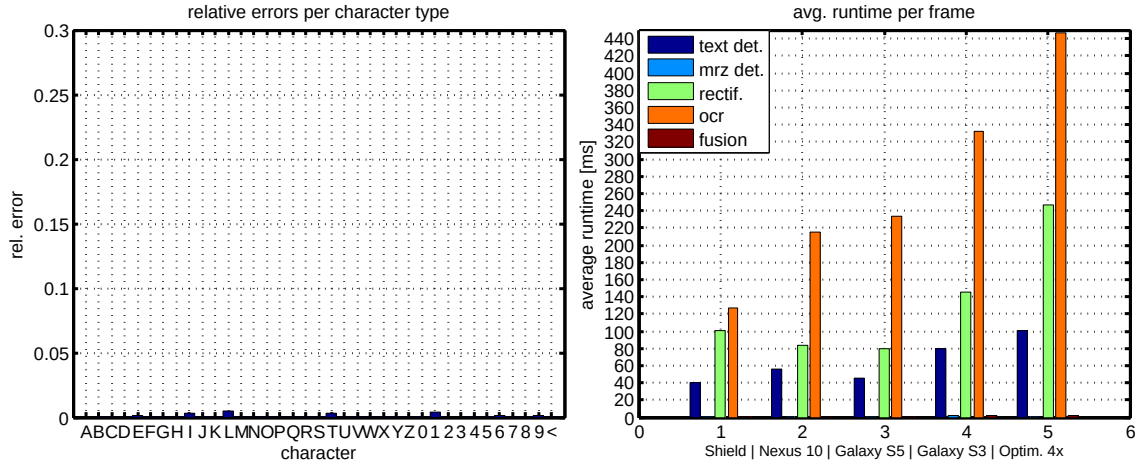

Figure 4.12: Left: Errors for individual characters (single-image database): In some cases, the characters I, L, T and 1 are confused with others. Right: Runtime of the prototype for various mobile devices (Android; subset of images): Runtime is dominated by patch warping and optical character recognition. The duration for MRZ detection from text groups and fusion is negligible.

4.5 Discussion and Future Work

We presented an approach for real-time MRZ detection and reading, which does not require accurate alignment of the document or the MRZ. By initial MRZ detection and fusion of results from several input frames, our custom OCR stage produces reasonable character

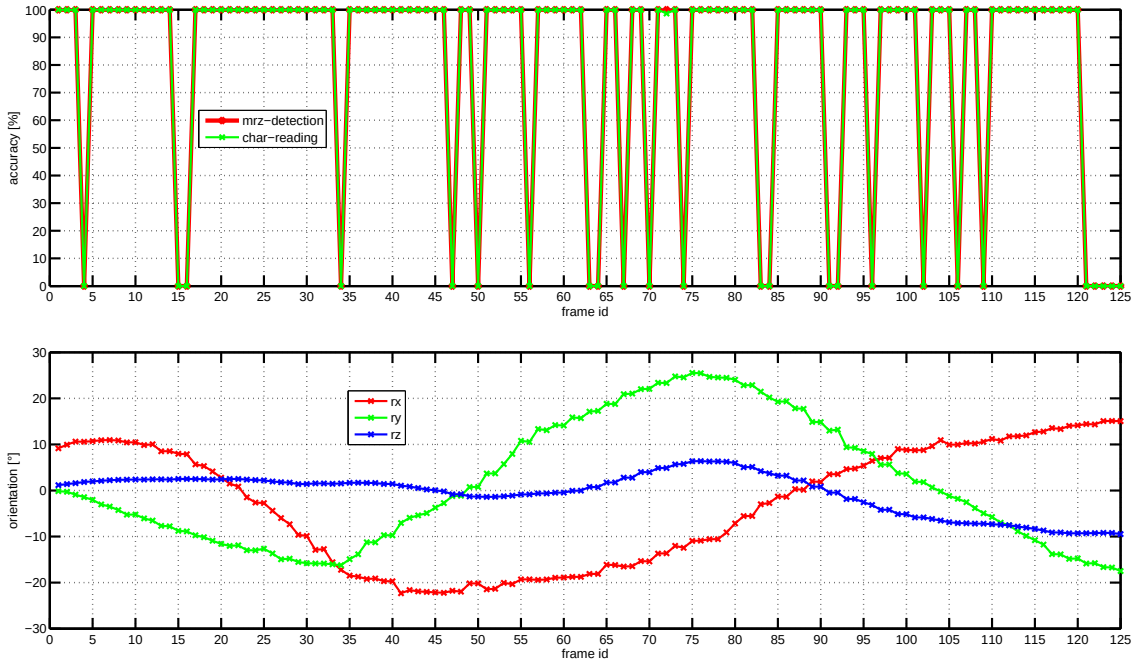

Figure 4.13: Top: Exemplary result of processing an entire image sequence of the synthetic database. The maximum gap size is two frames, which corresponds to a waiting time of less than 0.1 s, until processable data arrives (assuming a framerate of 30 FPS). Bottom: Corresponding orientation throughout the exemplary image sequence. The example document is captured from viewpoints that differ considerably from the ideal setting.

reading results despite having to deal with unaligned input. For evaluation purposes, we introduced a new synthetic database, which covers many different document backgrounds, MRZ contents and viewpoints. Saving the time required for alignment, MRZ data can be extracted faster than with state-off-the-art mobile applications. Based on the results of our experimental evaluation, some individual aspects deserve further discussion.

MRZ Detection: Detection from a single frame is difficult, as it might fail, if the document is viewed under steep angles. The overall MRZ recognition process clearly benefits from using a continuous video feed (see Figure 4.11). Due to the efficiency of our approach, frames can be processed in real-time, and instant feedback can be given to the user. Due to the larger amount of data, missing single frames is not critical.

Character Recognition: Although reasonable character recognition rates (exceeding 95%) could be obtained during our evaluation, a closer inspection reveals that in some cases, the current prototype confuses the characters I, L, T and 1 with similar samples (see Figure 4.12). Beside character confusion, occasional issues in character segmentation make up most of the remaining cases due to region splits. This could be improved by a machine-learning approach on the extracted patches (e.g., SVM). It is important to

note that for full MRZ reading, a heavily tuned character recognition engine has to be employed, suffering from a failure rate of at most $1e^{-4}\%$. Given the fact that real-world samples are hardly to be found in large quantities, this turns out to be a challenging problem on its own.

Image Resolution: We found that using a video stream with higher resolution (i.e., Full HD) in our mobile prototype only gives small improvements over fusing multiple frames with lower resolution, as proposed in this work. When processing such a stream on Android, there is noticeable latency, even though the full resolution is only used in the OCR stage. Due to this delay, there can be a lot of change between subsequent frames, causing occasional blur depending on user behavior. Since this is particularly undesirable regarding usability, it seems reasonable to stick with low or medium resolution images, employ an advanced frame selection strategy (e.g., depending on sharpness or lighting) and to further improve the OCR stage.

Future Work: It could be worthwhile to investigate the fusion of character segmentation results, instead of character classification results. This could help to further save runtime, since the time-consuming OCR stage would need to be evaluated only once for the final masks.

If more character training data becomes available, the template matching could be replaced with a suitable classifier. This would certainly help to improve full MRZ reading results including runtime. Our aim is to create synthetic character samples with different kinds of noise and other distortions in order to mimic all kinds of acquisition conditions and settings. Then, different machine learning techniques can be employed to improve upon the current approach.

The MRZ should be continuously tracked in order to support partial readings and to improve the monitoring of capture conditions. This would help to cope with temporary distortions like highlights which may currently prevent successful MRZ detection and reading. Then, really extreme poses and differences in the depth of field can be detected and rejected. For practical reasons, slightly curved documents should also be handled.

With a robust estimation of a transformation from the MRZ, the detection of document borders (see Chapter 3) could be improved. Since the corresponding edge-map can be fully rectified, searching for lines is greatly simplified. In addition, the search only needs to be carried out at a certain distance from the MRZ (dependent on the direction), since the position of the MRZ on the document is constrained by the available specification [74].

5 Hologram Detection and Verification

Contents

View-dependent elements such as holograms are used frequently on security documents. They change their appearance depending on the viewing direction and the position of light sources in the environment. This property makes them an interesting element to be put onto all kinds of documents or goods for the purpose of ensuring originality.

Hologram verification requires that knowledge about both the existence and the visual appearance of such an element is given. In the Mobile AR pipeline proposed in this work, such knowledge can be provided by the system, once the class of the document has been identified. Since knowledge about the presence of a hologram in a document can also be beneficial for document classification, it is interesting to investigate the feasibility of performing automatic hologram detection.

However, it is a common practice to substitute a hologram by a similar one[1]. In this case, detailed knowledge about the behavior of the element is required in order to reason about the validity of a document. Although such information can be provided by a document information system after successful classification, due to the dependence of the element onto the viewing conditions (i.e., light sources in the environment) it may still involve undirected movements. This does not assure true correspondence of the viewing direction and the appearance of the element. With a Mobile AR system, such information is available and can be exploited.

[1] according to a domain expert consulted

5.1 Contribution

We investigate the feasibility of hologram detection on off-the-shelf mobile devices, when no prior information about the contents of a document is given (see Section 5.2). This mimics the case of reference-free verification, as performed by untrained individuals. Using efficient algorithms suitable for mobile application, the location of one or more holograms can be determined using a series of registered images taken automatically during changes in orientation of the device or the document, caused by the user [59].

We also investigate the feasibility of hologram verification using off-the-shelf mobile devices by proposing a setup for repeatable image capture of hologram patches [62]. By using the built-in flashlight of a mobile device and by processing images within a mobile AR framework, data suitable for visual comparison of hologram patches can be captured. We further propose a setup for capturing reference information for comparison and investigate the possibility of performing automatic matching on the mobile device.

To the best of our knowledge we are the first to tackle these tasks in the context of off-the-shelf mobile devices.

5.2 Feasibility of Mobile Hologram Detection

The main contribution of the work presented in the following is the automatic detection of both the presence and location of holograms on a security document using a Mobile AR setup. This has multiple use cases, such as the detection of document layout for a subsequent classification step or automatic model building including verification.

The necessity of sampling the appearance of holograms from multiple view points is not apparent to the naive user. Therefore, as a side contribution, we propose an AR game concept which causes the player to capture the appearance of the document playfully, without the need to consider details about the nature of holograms. The results of a user study proof the plausibility of this approach.

5.2.1 Document Detection, Tracking and Registration

Security documents are usually made of paper or cardboard and are generally of rectangular shape. For reasons of robustness and efficiency, we limit ourselves to roughly planar regions.

5.2.1.1 Detection and Tracking

We first generate a suitable document template which can be used for frame-to-frame tracking or a dedicated registration step. It is based on an algorithm for the detection and rectification of perspectively distorted rectangles, serving as tracking targets (see Chapter 3). The user is asked to place the device in front of a document and to trigger the detection. Once the document region has been detected, it is rectified and stored. The

rectified image is then used to create a planar tracking template represented as an image pyramid at runtime, which can be tracked using natural features [174]. Harris corners and NCC are used to match patches across subsequent frames and to establish a homography between the current observation and the rectified target. A motion model is employed to estimate and predict the camera motion, saving a considerable amount of computational resources. As a result, the algorithm can be used in real-time on modern smartphone hardware, as it delivers a full 6DOF pose for each frame.

This setup has the advantage that is allows to interact with previously unseen documents having arbitrary personal data on it. In the context of subsequent CV algorithms, knowledge of the current viewpoint can be beneficial, as it allows us to work with rectified images and to control image capture.

5.2.1.2 Image Stack Creation

In order to decide upon the presence of holograms on a document, it needs to be recorded from several viewpoints. A minimum of $n = 2$ frames recorded from suitable viewing angles is required to be able to reason about the presence of such elements. For reasons of robustness, more viewpoints should be recorded. Based on the results of the target tracking module, our algorithm consists of three main parts to create an image stack: (i) frame selection, (ii) warping/registration and (iii) spatial filtering (see Figure 5.1).

Frame Selection: Ideally, the image stack should contain poses to cover the variability of a view-dependent element in the best possible way. This is not an easy task for inexperienced operators. Therefore, in favor of repeatability and reduced cognitive load, the task of frame selection is not assigned to the end user. We use the obtained tracking pose to automatically select frames based on a 2D-orientation map (polar/azimuthal angle) centered at the document (with some pointing tolerance) and also consider target visibility and template similarity.

Registration: For every frame which passes the selection step, the estimated homography from the tracker pose is used to create a rectified image. A full set of frames generates a stack of equally sized pictures (see Figure 5.2). In general, the tracking algorithm is rather robust and can track the target successfully over a wide range of viewing angles. Parts of the target may move out of the camera image, and the observations may undergo

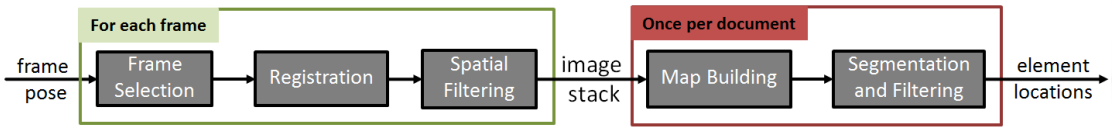

Figure 5.1: Illustration of the required steps for hologram detection per frame and per document/evaluation of the image stack.

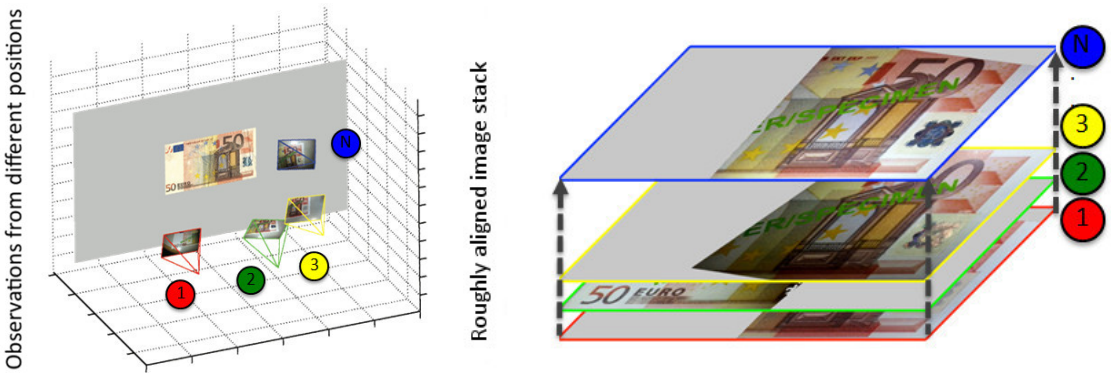

Figure 5.2: Left: The target is tracked and observations from different positions are recorded. Right: Through the estimated homography, each image is rectified and pushed onto the stack.

significant perspective distortion. The rectified images can therefore be incomplete or show alignment problems.

We experimented with refining the alignment in an additional step with feature extraction, windowed matching and homography estimation. However, this degrades the frame rate, which is not desirable from the standpoint of usability. As frames are constantly delivered by the camera, we, instead, chose to reject badly registered frames using NCC scoring, which is computationally cheaper. Due to the real-time tracking in the proposed setup, this is a reasonable way for automatically selecting usable frames.

Spatial Filtering: Each new layer added to the stack of registered images is spatially filtered to better cope with noise and remaining inaccuracies in registration. We use a windowed mean filter for this task, which is based on integral image computation [146]. We account for incomplete image information (black areas in warping) by recording valid areas used in filtering in a second mask.

5.2.2 Hologram Detection

Unlike other effects, such as specular highlights, visual changes caused by holograms remain spatially constant. Our approach is based on the idea of tracking changes in appearance over time, using the registered image stack as a starting point.

Our algorithm for processing the stack consists of two main parts: (i) map creation by statistics-based voting and (ii) a segmentation and mode-seeking algorithm for creating the final detection result (see Figure 5.1). An optional verification step is also added, which uses NCC computations at the estimated hologram positions throughout the registered stack to reject false positives.

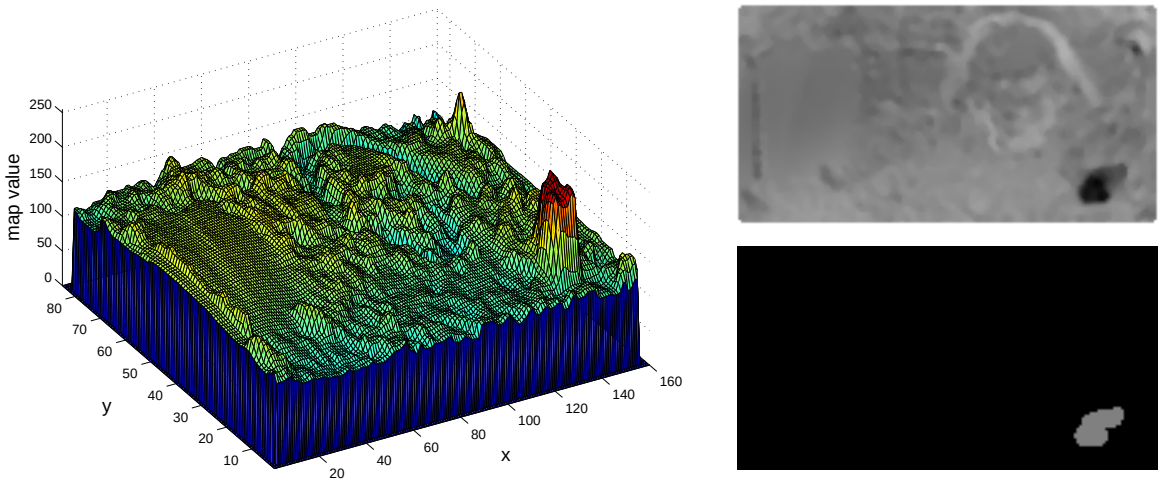

Figure 5.3: Left: Surface plot of a hologram map obtained from a sample document. Top-right: Corresponding scaled intensity image. Bottom-right: Selection result using adaptive thresholding.

5.2.2.1 Map Building

We treat the image stack at each position (x, y) as a series of measurements. We assess the amount of change by computing a suitable error concerning a model m at position (x, y) over the entire stack, obeying the masks computed in the previous step. This finally gives an intermediate representation of evidence for view-dependence, which we call *hologram map* (see Figure 5.3). We tested using the mean m_0 or the median m_1 along with the average quadratic error in image space

$$e_0(x,y) = \sqrt{\frac{1}{L(x,y)-1}\sum_{l=1}^{L(x,y)}(v_l(x,y)-m)^2} \tag{5.1}$$

or the average absolute error

$$e_1(x,y) = \frac{1}{L(x,y)}\sum_{l=1}^{L(x,y)}|v_l(x,y)-m|, \tag{5.2}$$

with $m \in \{m_0, m_1\}$ in different combinations, where $L(x, y)$ is the number of stack layers that contain valid entries for position (x, y) according to the obtained masks, and $v_l(x, y)$ is the pixel value in layer l.

In case of the pair m_0, e_0, model-building and error computation can be done on-line, which requires relatively few computational resources.

5.2.2.2 Segmentation and Filtering

We seek to localize dominant spatial peaks within the hologram map and the adjacent regions of large changes of similar magnitude. Consequently, this can be treated as a segmentation problem, where the choice of the method influences both quality and runtime. As the content of the map is highly dependent on the nature of the document itself, it is not sufficient to just apply a global threshold. In contrast, we use locally computed thresholds, which are additionally adapted using global information [10] (see Figure 5.3 for an exemplary result). In order to save runtime, integral images are used for filtering.

The computed regions are filtered in order to reduce false positives. We use minimum area, aspect ratio, and compactness along with a minimum magnitude/homogeneity criterion on the obtained region.

5.2.3 Experiments

We recorded several documents with holograms using a Samsung Galaxy S3 smartphone (Quad-Core ARM Cortex A9 CPU, 1 GB RAM, Mali-400 GPU, Android 4.1.2) with and without flashlight enabled. We used Euro banknotes and several samples, mainly attached to prints of specimen documents, giving a total number of 14 different holograms.

We quantized the orientation map with a step-size of 2 degrees and limit the extension to 25 degrees in each direction. We aimed for relatively high number of frames (90) in order to allow a more detailed evaluation of the algorithm. However, this is not a problem due to the real-time tracking. Document regions are warped to have a maximum extension of 160 pixels, and spatial filtering is carried out on a 3x3 window.

The initial workflow consisted of detecting the document and moving the phone or the document around, logging image frames and pose data to memory. This data is fed into our algorithm and analyzed concerning accuracy. The most promising setup is timed directly on the mobile device. In order to obtain more representative results, we captured each document three times, with and without flashlight.

The user needs to capture the document from various viewpoints in order to gather the required data for the algorithm at hand. Since it is not obvious, how the user can be supported during this process, we exploit an existing context for this task an evaluate it within a user study.

5.2.3.1 Accuracy

In order to get reasonable insight concerning the performance of the algorithm, we manually annotated all template images, producing reference masks for hologram occurrence and location. These are then considered as ground-truth for the remainder of the task.

Our scoring is based on a layout distance metric originating from document retrieval [168]. Similar to layout distance metrics for documents, our metric has to account for missing or superfluous elements, but we only consider overlapping regions (e.g., without

Figure 5.4: Left: Using Google Glass to perform hologram detection on a foreign passport. The picture as seen by the user is depicted in the inlay in the lower right corner. Right: Visualization of exemplary detection results in our prototype. Note that this is based on an approximation of the hologram region by a bounding rectangle, although we obtain a more detailed estimate of the region.

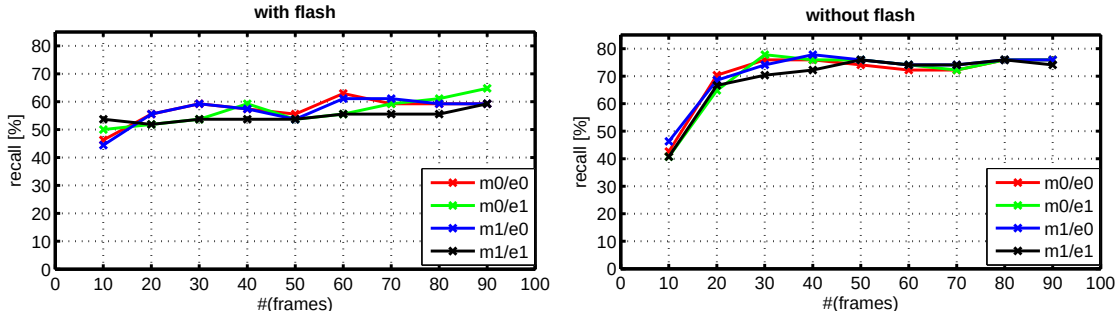

Figure 5.5: Results of hologram detection using 14 different documents when, binarizing the hologram map with flash enabled (left) and flash disabled (right). Not using the flashlight gives better results, since specular highlights can hamper map evaluation. There are only small changes, when more than 30 frames are used.

Manhattan metric), relating a region present in the ground-truth mask R_{gt} to the regions R_j obtained by our detection approach (see Equation 5.3), where Ov denotes the number of overlapping pixels and s_{ov} the obtained score. We treat a detection as true positive, if $s_{ov} \geq 0.4$. In case several regions in the detection result have a sufficiently high overlap with the same ground truth region, the best one is counted as true positive, whereas the others are regarded as false positives.

$$s_{ov}(R_{gt}, R_j) = \frac{2Ov(R_{gt}, R_j)}{area(R_{gt}) + area(R_j)} \tag{5.3}$$

We ran the proposed hologram detection algorithms using different algorithmic combinations for analyzing the image stack (see Figure 5.5). Obviously, there is little difference concerning accuracy for the evaluated methods. From around 30-40 processed frames upwards, there are only small changes when adding more frames. Best results regarding

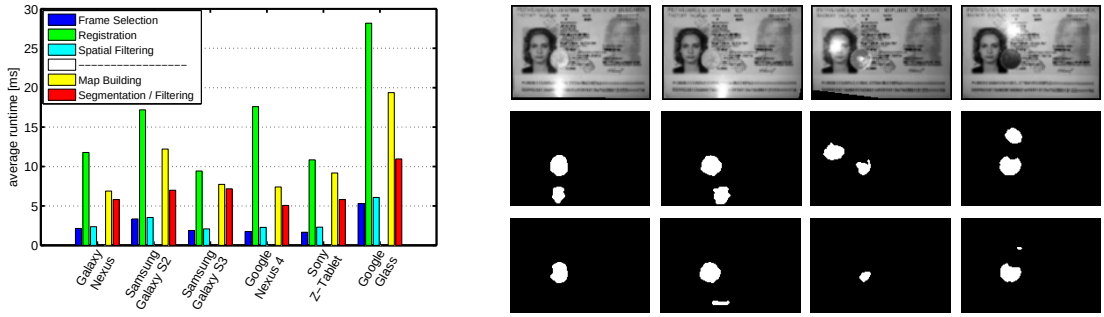

Figure 5.6: Left: Runtime for the individual parts of our approach on several different devices. Note that the first group of tasks is done once per frame, while the second group of algorithms needs only be done once per session. Right: Segmentation of single stack layers. Input image (top row), MSER regions (middle row), MSER regions from modified input image (highlight detection, inpainting)

practical applicability are obtained by using the combination (m_0, e_0), giving a recall of ~ 0.75. We omit plotting precision, since it was found to be at maximum value in almost all cases.

Interestingly, using the flashlight gives worse detection results. We found that this is often due to specular highlights, which make map evaluation considerably harder.

5.2.3.2 Runtime

Runtime of the proposed algorithm can be divided into two overall parts. The first part, building and updating the image stack, needs to be done on a per-frame basis and, therefore, needs to be very fast. The second part, the final evaluation of the hologram map along with the subsequent validation step, is done at the end of the capture operation. Therefore, this step is less critical concerning runtime. We made experiments on several different mobile devices employing a representative subset of documents using the most promising setup determined during evaluation of accuracy, (m_0, e_0).

According to Figure 5.6, the individual algorithmic parts take between 13-25 ms per-frame on most devices. Obviously, warping using the available pose information requires most of the time. Final evaluation of the hologram map including segmentation and filtering takes between 13-31 ms, but this needs to be done only once per session. Overall, the Samsung Galaxy S3 is the fastest device per frame, whereas the Nexus 4 is the fastest one for generation and evaluation of the map. Google Glass is the slowest among the tested devices, taking around 40 ms per frame. Our Glass developed a relatively hot surface temperature during testing, which quickly causes thermal throttling.

It must be noted that the aforementioned optional validation step of a detected region takes several hundred ms. However, it only needs to be done once per session. Upon successful detection, we augment a semi-transparent shape at the corresponding positions (see Figure 5.4).

5.2.3.3 User Guidance

Various movements of a mobile device over a surface are common with mobile AR games. In the following, we investigate if such a setting can be useful for hologram detection.

Concept: We crafted a mobile AR game which can be played on an arbitrary planar document. The goal is to help a hungry eagle flying around the document to get hold of suitable prey (spider, snake, cobra), which is sitting on a regular grid augmented onto the document (search mode). Mimicking the eagle's eye, we use a sighting device and perform loose matching of the center and viewing direction. Cells (base point), polar angle and azimuthal angle of the ray are randomly selected for placing prey (see Figure 5.7). This means, several randomly sampled hemispherical subspaces are used.

Upon successful alignment, the eagle dives towards the prey to get hold of it (attack mode), triggering an attack animation of the prey. Then, the eagle flies away (carry mode), while the user receives a score (see Figure 5.8).

User Study: We evaluated the game running the proposed hologram detector in the background within a user study taking place in several offices around our institute. Participants (11, 1 female) were briefly introduced to the game concept and mechanics on an off-the-shelf mobile phone. They were asked to play the game on a document used in our previous evaluation, logging task completion time, score and hologram detection results onto the flash memory of the device. The participants' opinion regarding enjoyment, motivation and the game's usability was recorded in a questionnaire along with user comments right after gameplay. It must be noted that participants did not receive any hints on the existence or purpose of the background task. Consequently, the progress bar within the app was modified to just count the number of gaming rounds, instead of the frame acquisitions by the detector. The end of gameplay is either triggered by the background task (detection algorithm finished) or by the game (maximum number of rounds (32)).

The data captured by 7 of 10 users during gameplay was suitable for hologram de-

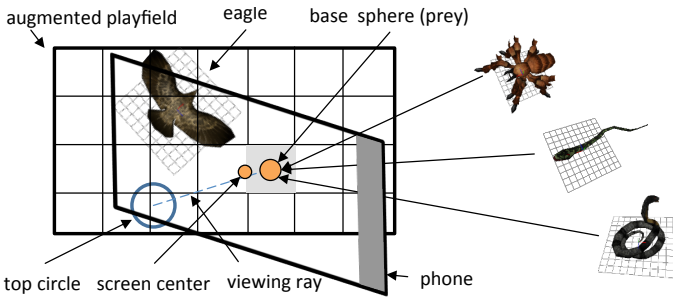

Figure 5.7: Mobile AR game concept: Cells are augmented on the document for prey placement. The sighting device consists of a base sphere, top circle and a focus point on the screen. For alignment (triggering the eagle to dive for prey), all elements must roughly coincide.

tection. Additionally, another suitable hologram map was recorded, which did not make it through final verification ($M = 0.75, SD = 0.40$). Participants played for around two minutes ($M = 98.9s, SD = 46.6s$). Gameplay was generally stopped by the background task except in one case, where the participant had severe issues with the interface. Due to the unusually high completion time (231 s), we treated this participant as an outlier.

All questions related to user experience were rated on a 5-item Likert scale ([-2,2] interval). Generally, users enjoyed the game ($M = 1.40, SD = 0.48$) and felt motivated ($M = 1.10, SD = 0.53$). The game was rated to be easy to use ($M = 1.30, SD = 0.90$) and to have satisfying controls ($M = 1.20, SD = 0.60$). The prototype was specifically described by five users as being 'fun'. However, three users reported on the repetitive nature of it. One user suggested to build several mini games or to increase the challenge by requiring finer alignment depending on the type of prey. One user reported that the prototype was difficult to use at the beginning, but increasingly better when progressing.

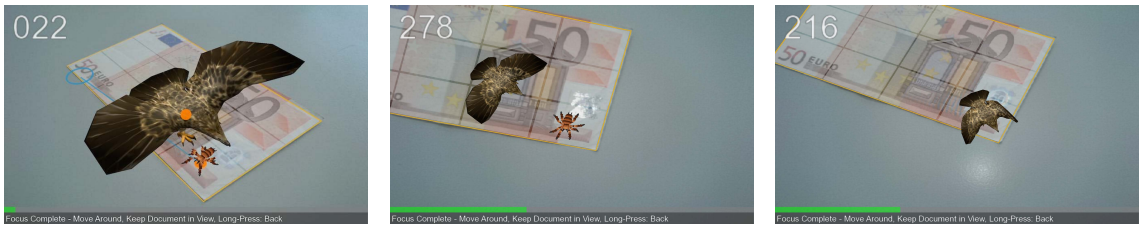

Figure 5.8: Screenshots from the game taken on the mobile phone. The eagle is circling over the target, on which the system places animated prey. By placing prey it suggests a new viewing position the user should reach with the mobile device to make the eagle attack and score.

5.2.4 Discussion

The proposed approach has notable similarities with existing background subtraction techniques. However, we use real-time tracking to obtain registered images from a number of viewpoints and only segment the final map to obtain candidate regions, which are subsequently validated. All this effort results in a pipeline that can be readily integrated into existing applications for document verification, as it delivers reasonable results at a very small runtime overhead during interaction.

Although our algorithm performs reasonably well on many public security documents, very shiny surfaces and holograms with a small number of different appearances cause false positives/negatives. We went to tackle these problems by using different map segmentation methods like MSER [107] or Mean-Shift [32]. However, in particular for Mean-Shift, this lead to less consistent results with serious region fragmentation, especially for more complicated backgrounds or very challenging lighting conditions.

In order to tackle reflections, we added a highlight detector and performed inpainting, which seems to improve results (see Figure 5.6). It seems more reasonable to carry out some more elaborate analysis of the image stack, however. As preliminary tests showed,

this comes at the cost of a notable runtime overhead on mobile devices and requires more in-depth investigation in the future.

An evaluation of the detector during a mobile AR game showed that in 8 of 10 cases, suitable data for hologram detection could be captured in the background, although no hints were given to the user on the actual purpose of the evaluation. It seems that the limited number of randomly placed sampling positions only partially maps to the image acquisition task of the detector. Probably a feedback channel from the orientation map towards prey placement could improve results in this case. Some users criticized the repetitive nature of the game in its current state and gave suggestions for improvement. However, meeting the demand for finer alignment might be counterproductive for the speed of the game, which is a major source of enjoyment and motivation. Using several prey locations/fly paths simultaneously could lead to a more challenging experience, while keeping the game's main appeal. In addition, the game setting could be improved by using more graphical elements like a sandy document texture, additional animals interacting with the game events or even a nest for the eagle.

5.3 Feasibility of Mobile Hologram Verification

During hologram verification, a set of appearances must found on the current element and compared to reference information. If mobile devices should become useful for hologram verification, it must be possible to repeatably capture such appearances with the built-in camera. Consequently, the observed appearances must be similar w.r.t. given reference information despite changes in capture conditions and unexpected user behavior. In the following, we propose a suitable setting for recording holograms using off-the-shelf mobile devices and evaluate its feasibility to serve as input for automatic matching.

5.3.1 Recording Holograms for Mobile Verification

View-dependent security elements show high-detail images that change drastically depending both on the viewing direction and the dominant light direction. Therefore, a single image cannot capture the full appearance of such elements. We chose to represent the elements using a SVBRDF representation [57] that allows us to both preserve the dependence on viewing and lighting angles as well as the spatial variation of the images. Furthermore, we are only interested in planar, thin surfaces - printed documents. Therefore, we do not require accurate models of self-shadowing or subsurface scattering effects.

However, because we are targeting a handheld mobile application where the device and the document are both moving, we require a full BRDF representation, as opposed to a surface light field [77]. Thus we are effectively using a 6D appearance model per color channel, where the radiance I is a function of both location (x, y) on the document, as

well as incoming light direction l and viewing direction d:

$$I = I(x, y, l, d). \tag{5.4}$$

The direction vectors l and d are unit length and therefore have only 2 degrees of freedom.

We are mainly interested in showing a representative image of the view-dependent element to the user. Therefore, we make several simplifying assumptions. We assume that the total radiance from a point on the element is dominated by a single major light source direction. Thus, we do not integrate over all incoming light directions, but a single snapshot is enough, given the dominant direction. Furthermore, we do not require a fully radiometric calibration and do not control for automatic exposure and white balancing of the camera.

We simply sample the appearance as a set of images indexed by viewing direction d and light direction l. We do not attempt to estimate a smooth BRDF model covering all points on the element, but rather keep the individual images as the final representation. This preserves the sharp changes in appearance, when the element flips from one view to another, as well as the necessary detail in the spatial domain.

5.3.1.1 Light Source

In practice, the dominant light direction poses a challenge in a mobile setup. Without any prior knowledge, we cannot reliably index into the list of appearance images. Therefore, we use the LED light source on a mobile as a constant source of illumination in the scene. As this is usually close to the camera, it dominates other light sources in indoor scenarios. Because the LED is fixed with an offset vector o with respect to the camera, the light direction l is a function of the camera pose with respect to the document (see Figure 5.9). The light direction is now proportional to the camera position P plus offset vector o, rotated by the camera rotation in world coordinates.

$$l \propto P + R \cdot o \tag{5.5}$$

For a fixed distance to the surface, P is just a rotated vector, and we obtain a similar equation for the viewing direction

$$d \propto R \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{5.6}$$

Thus, our representation is reduced to a 5D model, indexed by the full 3D camera rotation and the location (x, y) on the document.

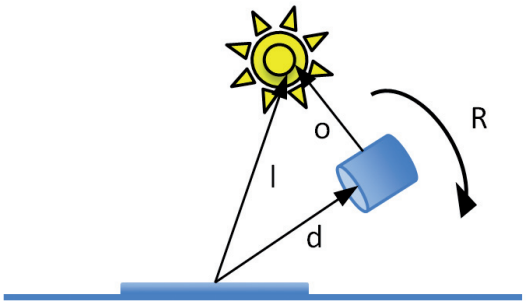

Figure 5.9: With the LED light source in a fixed configuration to the camera, there are only three degrees of freedom in the input to the SVBRDF function.

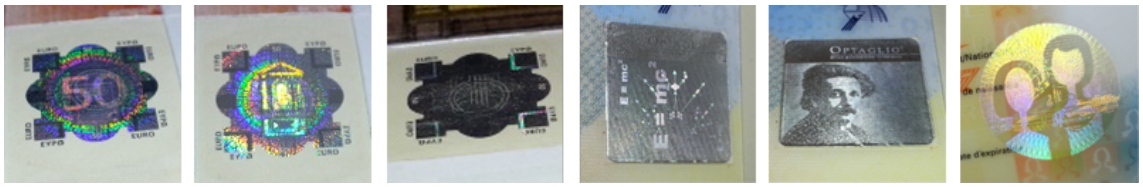

Figure 5.10: Captured appearances of view-dependent elements using the proposed setup.

5.3.1.2 Feasibility

We performed an initial feasibility check w.r.t. image capture on the Samsung Galaxy S3 mobile phone using the built-in camera and flashlight. We captured various holograms on banknotes and plastic cards and observed whether the recorded appearance matches those illustrated in given reference material for the element under consideration. According to Figure 5.10, it is possible to capture different appearance states of view-dependent elements with this setup.

When operated at a small distance to the document, the built-in LED flashlight dominates other light sources in typical indoor scenarios. This assumption is invalidated with strong artificial light sources or when operating outside (e.g., direct sunlight). In such cases, the workspace must be carefully shielded (e.g., manually).

The flashlight may introduce severe specular highlights, even directly on the hologram. These highlights usually appear around the orthogonal view of the target, but do not affect the application much, because the more interesting views for verification are often at an angle away from the normal. Moreover, the verification of most holograms does not require dense sampling, but relies on a rather limited number of specific views.

The location of the LED close to the camera would indicate that the light direction l can be approximated with the viewing direction d. However, we tested this, and we clearly observed a dependency in the appearance on rotations of the device around the camera's optical axis. Figure 5.11 shows an example. While the phone was pointed along the same direction from the element, it was rotated around the optical axis, leading to different images.

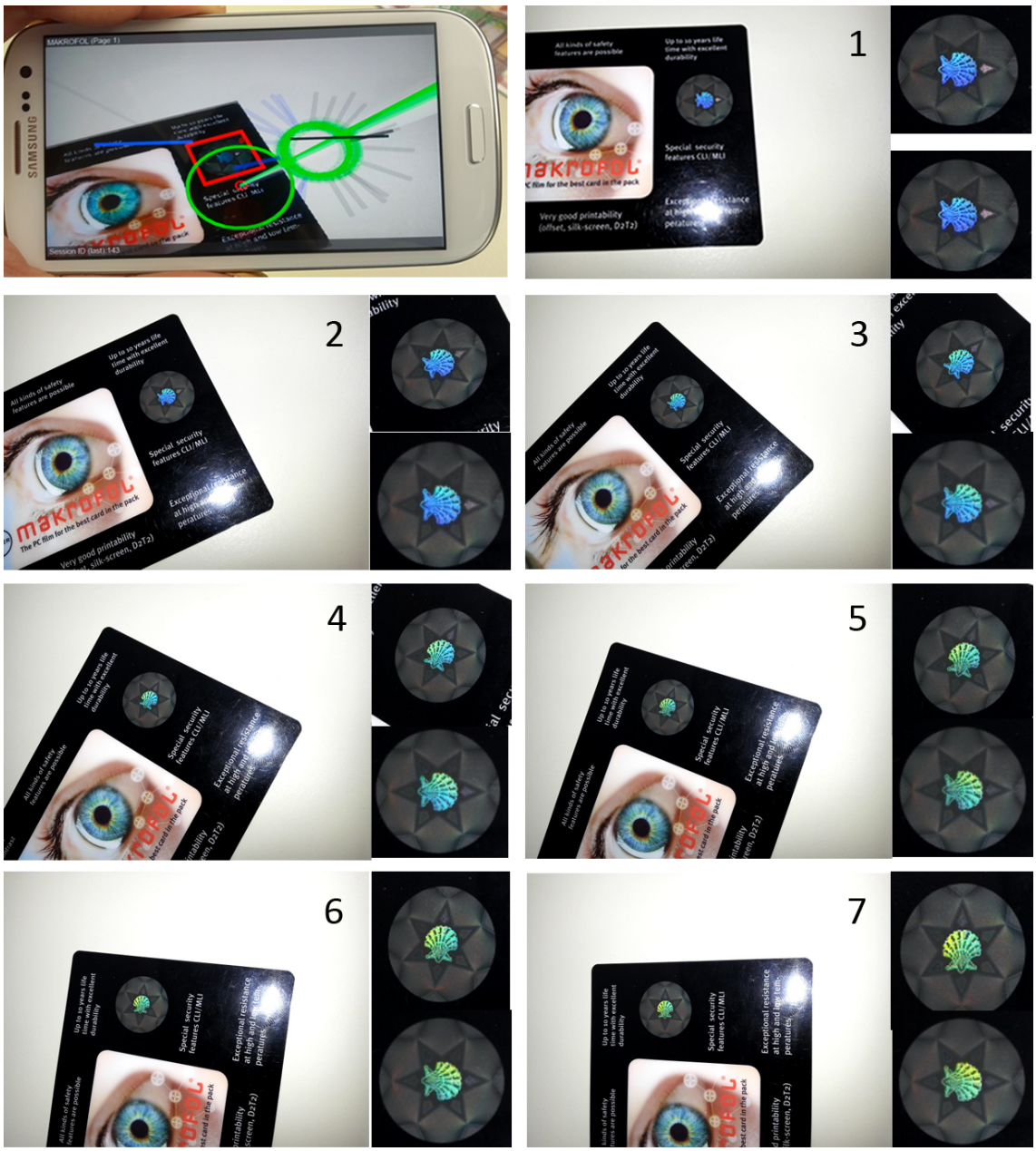

Figure 5.11: Visualization of repeated capture while rotating around the optical axis (top left). Dependency of rotation around the optical axis on the appearance (images 1-7). The upper patches show element images captured by the camera. The lower patches show the rectified element images that form the appearance model.

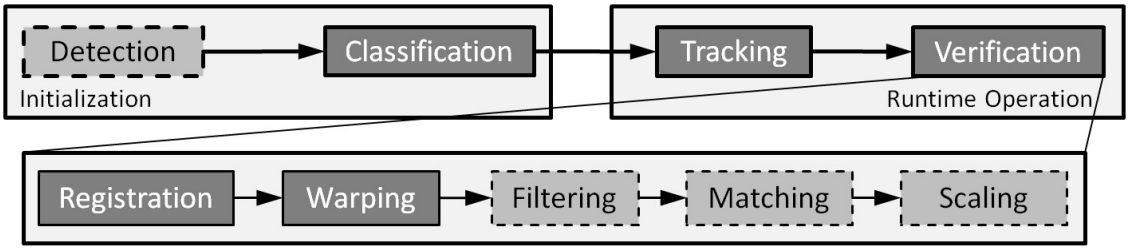

Figure 5.12: Overview of our mobile hologram verification pipeline. For manual verification, images are registered and the extracted hologram patches are rectified. In case of semi-automatic verification, additional processing is required (dotted rectangles).

5.3.2 Framework for Mobile Hologram Verification

Mobile hologram inspection relies on a pipeline performing document detection/classification and tracking/verification (see Figure 5.12). Assuming suitable reference information, the position of holograms on the document template is known, once the classification step is completed. During the capturing process, the tracking pose allows to assess the correct viewing angle and to rectify the target region from the video frame for visualization and matching. This approach does not need any further user input, besides covering all possible viewing directions and orientations.

5.3.2.1 Basic System

Following the insights gained from feasibility testing, we constructed a mobile AR prototype for hologram verification. By visually tracking the known document, the system estimates the current viewing direction and camera pose. Available layout information can be represented by an initial augmentation, providing instant feedback on the presence and location of relevant security features (see Figure 5.13).

Both detection and tracking rely on the assumption that we are observing planar objects. This is often violated with paper-documents, however. In most cases, this does not lead to tracking failure, but pose jitter. We smooth out the poses in a ring buffer to improve stability. Averaging the pose over 2-3 frames stabilizes the view, while the introduced lag is small for this particular setup.

Viewpoint and flashlight act as triggers for different appearances. From our experience, not using the flashlight as a dominant light source does not give repeatable results. According to informal tests, there is limited invariance with different devices, depending on camera properties, flash intensity and relative position (see Section 5.3.3). Consequently, an initial probing of the current lighting conditions is required before the actual verification task can take place. When the exposure can be fixed on the mobile device, this can be achieved by activation of the flashlight and thresholding the relative amount of saturated pixels in order to reason about the dominance of the built-in light source.

Figure 5.13: Mobile document verification system tracking a sample instance of the data page used in passport. The position of security features is augmented directly onto the document (left). Detailed information about an element can be triggered by pointing the camera at it (right).

5.3.2.2 Selection of Reference Data

Hologram verification requires a set of reference appearances of the corresponding region, which needs to be matched against the captured image data at runtime. Consequently, the first step is to select a series of viewing directions relevant to the verification process.

The choice of reference data for verification depends on the hologram (e.g., number of transitions) and is constrained by the particular setup being used. Normally, the main source of reference information is a printed manual, which is obviously missing the corresponding pose information. However, the verification of holograms in the context of mobile AR poses additional constraints. For each viewing direction, we require stable tracking and reproducible appearance. While the former mainly excludes low angles and extreme close-up views from being recorded (tracking failure), the latter limits the maximum viewing distance and avoids orthogonal angles, which produce specular highlights due to the placement of the LED light. In practice, it seems reasonable to operate roughly at constant distance from the hologram, giving a hemispherical capture space. We consider two views to be the minimum for verification of view-dependent elements.

5.3.3 Systematic Recording and Automatic Matching

An automatic way of sampling reference data from a given hologram is desirable for reasons of temporal effort and accuracy. In the following, we describe a setup for systematic image capture, which provides a reliable basis for the selection of reference data. This data can then be used for the actual verification step taking place at runtime.

Hologram verification using Mobile AR has the advantage that the hologram is captured under the right viewing angle and lighting conditions. Then, visual inspection can be carried out by the operator, which is the same as with a printed or digital manual. However, it is reasonable to assume that an automatic matching step will be more efficient, since it allows come up with a final decision by the system without requiring further user intervention.

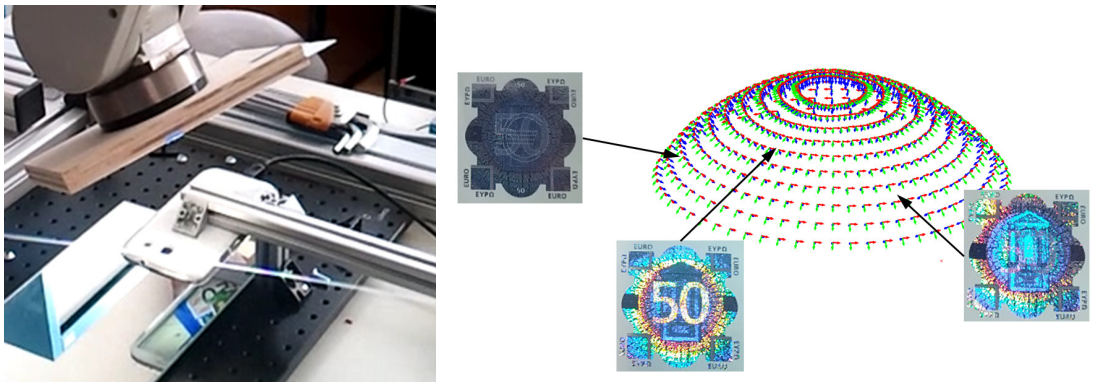

Figure 5.14: Left: We sample the view-dependent element on a document using an industrial robot and an off-the-shelf mobile phone. Right: Due to using a single dominant light source, the element is sampled from viewpoints situated on a hemisphere.

5.3.3.1 Systematic Recording

With moderate ambient light, the appearance of a hologram is largely dominated by the LED flashlight of mobile devices. This essentially means that the workspace consists of a hemisphere centered at the hologram on the document. This space must be further restricted due to the nature of the light source (highlights on orthogonal views), the tracking system (robustness with low angles) and the operator. We use an industrial robot (Mitsubishi MELFA) for capturing all relevant appearances of a view-dependent element (see Figure 5.14). This allows reliable sampling of holograms and eliminates undesired human influence. We spatially sample a hemispherical space using the robot and remotely control the device. Initially, an orthogonal position is approached and an auto-focus operation is triggered on the mobile device, which results in a reasonable focus setting for the remainder of the process. We capture the current video image and the corresponding pose for each position on the hemisphere.

We assume the hologram to be planar and project its bounding box into the image using the recorded pose. We estimate an image transformation with respect to the hologram region on the undistorted template and warp the sub-image containing the hologram. For increased accuracy, we perform an additional registration step using the template of the document, before extraction and rectification of the corresponding patch. The result is a stack of registered image patches that represent all observable appearances of the current hologram. From all the recorded information, a relevant subset must be selected according to the criteria given in Section 5.3.2.2.

5.3.3.2 Automatic Matching

The automatic matching of selected reference information can be carried out on rectified patch data by using layout and pose information. The matching step itself demands a suitable similarity measure. In the following we elaborate on the usefulness of different

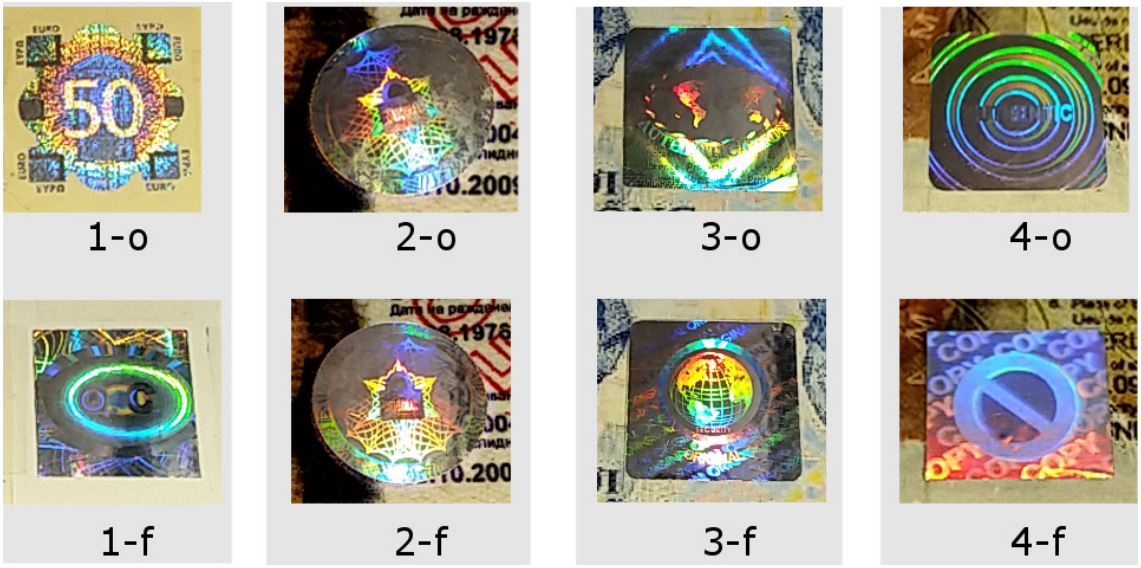

Figure 5.15: Holograms used in our matching experiments. Top Row: Original elements. Bottom Row: Substitutes

similarity measures for the purpose of mobile hologram verification.

Similarity Measure: A suitable similarity measure should be able to quantify the difference in appearance between genuine and fake holograms. Ideally, it should be invariant regarding variations in image capture conditions and the type of the hologram. In order to allow a subsequent inspection of the result by a human operator, such a measure should largely correspond to human perception.

In the following we evaluate a series of similarity measures on holograms recorded with the Samsung Galaxy S5 smartphone (see Figure 5.15). In each case an original, a copy and a substitute were recorded under typical office conditions using the built-in flashlight as a dominant light-source. The settings for the recordings included an office room with light switched off, fluorescent light and the aisle in front of it, which has more daylight influence.

We evaluated several similarity measures such as Sum of Absolute Differences (SAD) and NCC, which are often used for stereo matching [71]. Due to the requirement of correspondence regarding human perception, Structural Similarity Index (SSIM)[184] and Edge-based Structural Similarity (ESSIM)[28] are also included. Additionally, we evaluate SSIM with color patches by reporting the minimum value over all channels (CSSIM).

Evaluation is carried out as a binary classification task on each reference view of every hologram. The task is to assign the correct class to each recorded patch from an original, copied or substitute hologram based on pair-wise matching. The required matching thresholds are selected automatically, based on the difference in scores between original and fake patches for each reference viewing direction. The recording position is

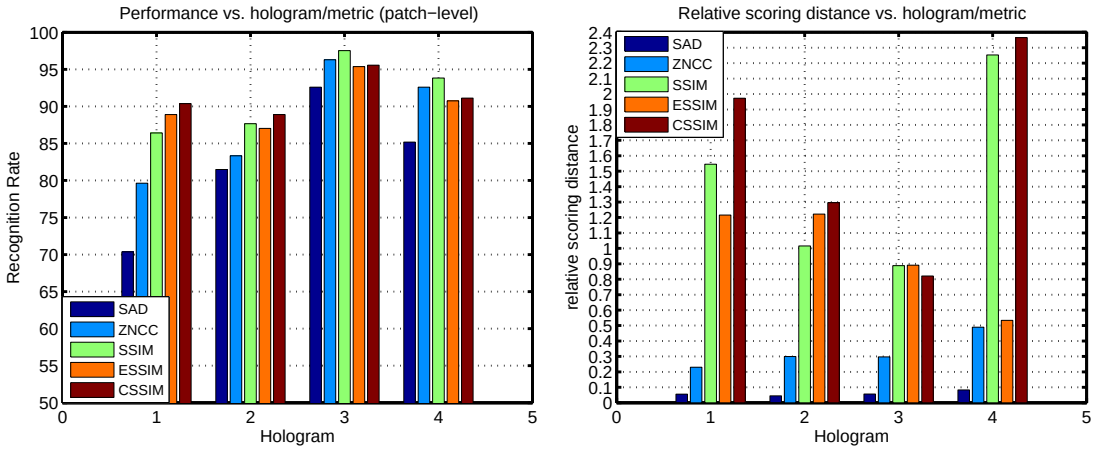

Figure 5.16: Left: Performance of various similarity metrics in recognizing fake and original patches when using data from all slices of the orientation map. Right: Relative difference of scores between fakes and originals. Note that the data was taken under various office conditions (no artificial light, fluorescent light, slight daylight (aisle)).

considered by matching patches only if the pose lies within a certain orientation threshold concerning the reference pose. Neglecting this relationship is not desirable, since elements having the same appearances at different viewing positions could not be differentiated anymore. This would weaken the security of the proposed approach.

There are notable differences in patch recognition regarding the type of hologram, but also the associated metric (see Figure 5.16). SSIM-based metrics, in general, give better results than NCC and, in particular, SAD. Overall, SSIM is stable, giving patch recognition rates of over 90% for holograms 3 and 4 and over 85% for the remaining ones. Hologram 1 is obviously most difficult to recognize regarding its originality (low SAD and NCC scores). We may speculate that this is due to the large amount of rainbow colors present on its patches.

For robust matching, the margin of the classifier should be as large as possible. Mapping this to the current task, the relative difference of matching scores between originals and fakes should also be large. Normalized relative scoring distances depict considerable differences between the evaluated similarity measures, but, also, between different holograms (see Figure 5.16). SAD and NCC only span a very small range compared to SSIM-based measures. So, it is much more difficult to set a reasonable matching threshold for them than it is for SSIM. Based on these insights, it seems more promising to use SSIM for matching hologram patches, instead of NCC. This is further backed up by results obtained from performing hologram verification using majority voting on individual patch matching results (see Figure 5.17). In this case, only SSIM allows to correctly recognize originals and fakes under typical office conditions.

We further investigated the patch matching performance for originals and fakes under optimal office conditions, when using different devices at runtime (see Figure 5.17). We

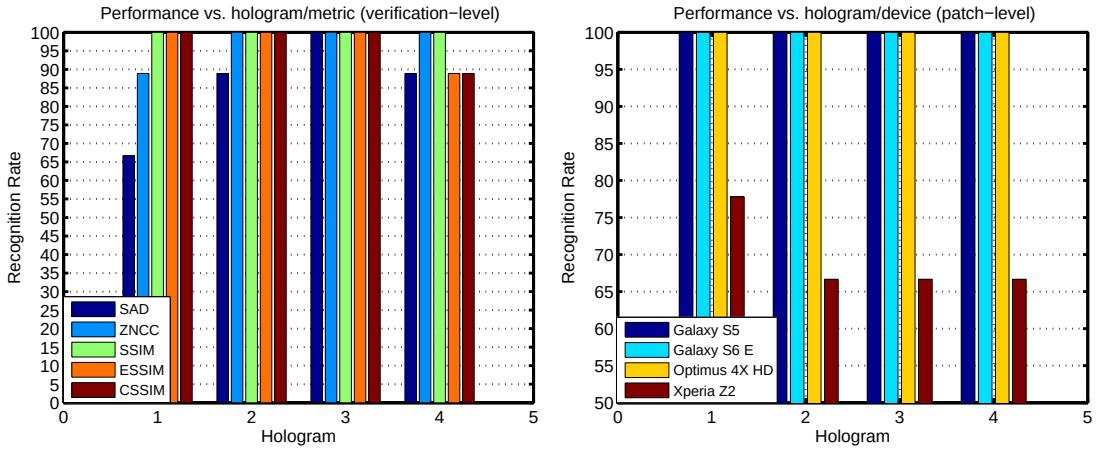

Figure 5.17: Left: Performance in recognizing fake and original holograms by majority voting on patch matching results under various office conditions. Right: Performance in recognizing fake and original patches when using various devices under optimal office conditions.

observed stable verification performance for two off-the-shelf devices (Samsung Galaxy S6 Edge, LG Optimus 4X HD). However, the Xperia Z2 smartphone failed. Further investigation revealed a relatively weak LED-light-source coupled with very different sensing characteristics. This device is not able to reproduce different appearances of hologram patches and cannot be used for hologram verification.

During feasibility testing we found that pre-filtering (Gauss or median filter) improves robustness in case of unstable views (e.g., rainbow hologram). Besides, additional shape-matching (e.g., modified Hausdorff distance [36]) can be used to reject false positives in case of very distinct patterns within the patch (e.g., stereogram). After computation of the matching score, linear scaling (k, d) of the result can be carried out based on coefficients computed from matching the captured reference data in an off-line step (see Figure 5.12).

5.3.4 Discussion

We observed that different appearances of holograms can be repeatably captured, if the built-in light source dominates the scene and if the hologram is recorded from the right viewing direction. We proposed to use a mobile AR framework for document verification for this task, since the available layout and pose information can be used for extracting rectified patches from the image. However, due to the constraints of this setup, it seems reasonable to support the user during image capture by an active guidance component (see Chapter 6).

The required reference information for comparison can be obtained by sampling the hemispherical space above the element and selecting a series of views according to several constraints imposed by the mobile setup. The user can then decide on the validity of the element by comparing the recorded data with reference information.

In order to improve the efficiency of the process, patch matching should be carried

out automatically instead of manual comparisons by the user. We evaluated different similarity measures for the purpose of hologram verification with originals, color copies and substitutes. The obtained results suggest that it is possible to automatically decide on the validity of appearances, but the results vary depending on the similarity measure used. NCC and in particular SSIM gave the most promising results. The actual choice is dependent on the hologram and the computational capabilities of the device. Ideally, the type of device should be the same for capturing reference information and on-line verification. Our experiments revealed that, for several devices, verification is still possible under optimal office conditions. While further invariance could be handled by using a machine-learning based approach for the comparison of patches, a reasonably large amount of training data is currently not available. Consequently, it lies in the responsibility of the actual implementation to detect the type of device and to retrieve the corresponding data for optimal matching performance. However, one popular device used in our evaluation cannot be used for hologram verification at all.

5.4 Conclusion

Detection: We presented an approach to automatically detect holograms on security documents with a mobile device. Our framework is capable of detecting and tracking a yet unseen document and automatically determining the location of one or more holograms, if present. For this purpose, a registered image stack is built in real-time and instantly analyzed, once the orientation space has been sufficiently covered. The experimental results presented proof of the plausibility of the algorithms proposed for use on off-the-shelf mobile devices.

In order to better detect holograms with a small number of views, a more elaborate stack analysis should be carried out. This is especially important for documents with shiny surfaces in case the flashlight must be used during image capture. In the context of practical application, it is also desirable to guide the user in order to efficiently record all the information required for detection.

Verification: We investigated the feasibility of capturing view-dependent elements using off-the-shelf mobile devices. Repeatable image capture is possible when using a dominant light source (i.e., built-in LED) and matching the required viewing direction. This has to be done for all relevant appearances of the element. Layout and tracking information, as available in a mobile AR framework for document verification, can be exploited to extract rectified patches for manual matching by the user. In order to better exploit the capabilities of a mobile setup, automatic matching should be performed by the system. An evaluation using different similarity measures with originals, fakes and substitutes revealed, that it seems realistic to build a semi-automatic system for hologram verification using an off-the-shelf mobile device.

It would also be possible to treat the matching step as a machine-learning problem and to decide on patch correspondence using a suitable classifier. However, this assumption demands a large number of real-world appearances of originals and fake holograms for training and testing, which are not available currently.

Due to the complexity of the task, it is reasonable to interactively guide the user throughout the process. An evaluation of our hologram detection algorithm within a mobile AR game gave encouraging results towards improving document security while playing. However, from the standpoint of practical usability, the efficiency of the process is a critical factor. Consequently, we will now focus on specialized goal-oriented user interfaces for the mobile inspection of holograms.

6 User Interfaces for Hologram Verification

Contents

The proposed setup for mobile hologram verification requires to record the element from a series of viewpoints, while obeying additional constraints such as operating distance or pointing at the element, which is imposed by the hemispherical capture space. In order to get reliable results within a reasonable time span, the user should be supported throughout the image capture process and possibly also during the actual decision phase. We must make sure, that all interesting space gets efficiently covered during image capture, so that a reliable decision can be made by either the user or the system. Especially when targeting non-processional users, the type and the parametrization of the user interface becomes a critical factor for the task at hand.

In the following, we present a series of experiments on hologram verification with off-the-shelf mobile devices, involving several different user interfaces, supporting manual and automatic matching of original and fake holograms [61, 62].

6.1 Contribution

We propose an approach for capturing holograms through view alignment, which allows the user to compare the expected appearance of a view-dependent feature and the real observed one under the current viewing direction (see Section 6.2). Our evaluation shows that such a setup can be used for repeatable image capture of hologram patches and is

able to provide suitable data for comparing patches. Although it is possible to verify holograms with this setup, it requires considerable temporal, physical and mental effort.

With the goal of improving the efficiency of the process, we propose several different user interfaces for hologram verification on off-the-shelf mobile devices using automatic capture and matching (see Section 6.3). Building on previous results, the problem is once more treated as an alignment task but also investigated in the context of constrained navigation. This finally leads to the design of a hybrid user interface. We compare these interfaces in a user study involving original and modified elements, informing a detailed discussion on the usefulness of these interfaces. Our results indicate that there is a significant difference in capture time between interfaces but that users do not prefer the fastest interface and are able to give better decisions on validity.

Following the insights gained from the aforementioned experiments, an alternative parametrization is proposed, which is mainly motivated by the analysis of typical user behavior when recording holograms with mobile devices. A subsequent evaluation within a user study with original and substitute holograms revealed significant improvements in matching accuracy and task completion time (see Section 6.4). The obtained results show, that holograms can be captured and automatically assessed in around 15 s, leading to better decisions on validity than those provided by laypersons.

6.2 View Alignment

The user can be guided to capture a frame from the same viewing direction and under the same light direction as captured in the reference image set. Using the LED light of the mobile phone as a dominant light source, the task is simplified to aligning the current pose of the mobile phone operated by the user with several reference views. For this task we propose a novel visual guidance approach for view alignment (see Section 6.2.1). While a comparison from a single point of view can lead to rejection, acceptance requires to check several viewing directions that present different appearances.

We investigated if it is possible to move to the correct viewpoints during a user study. Given pairs of reference and test images captured with a mobile device, we subsequently investigated patch similarity, but also user decisions on validity, and compared it to a digital manual based approach (see Section 6.2.3).

6.2.1 Conceptual Approach

We propose a visual guidance approach inspired by two widely known metaphors, namely iron sights and virtual horizon. Iron sights are used to align the viewing direction of the operator with the direction of the device. In general, shaped alignment markers are used for this task, which are positioned at a given distance on the device. Accounting for distance or scale depends on the task and requires a calibration procedure. This is often applied in sighting mechanisms.

The virtual horizon is an indicator of level, which is often used when a device needs to be aligned relative to the ground. At any time, the instrument shows the level of the object relative to earth gravity. Implementations range from a simple water level for mechanical tasks to advanced electronic devices used in aircrafts.

Based on these techniques, we subdivide view alignment into three steps: We match the direction of the viewing ray (iron sights), the position along ray and the in-plane rotation (virtual horizon). It is crucial to guide the user through these steps, so that accurate alignment can take place (see Figure 6.1 for a conceptual overview).

6.2.2 Implementation

We implemented the proposed guidance approach in an interactive prototype for mobile devices. The camera rotation of the tracking pose indexes into the stack of reference images for a given hologram. The reference image is brought up for comparison with an image captured from the live video frame.

The iron sights setup is realized by using two big circles, which mark start- and end-point of the viewing ray. By using the intrinsic parameters of the camera, we scale the lower ray circle so that it overlaps entirely with the top circle, once direction and distance match. For easier alignment, we additionally use a smaller ray base circle, which is intended to overlap with a small sphere fixed on the device screen. Their scale is also adapted with the intrinsic parameters. The virtual horizon setup consists of two lines placed at the top of the ray and two similar lines fixed on the screen. By using two different colors for each line, we account for a possible ambiguity in rotation around the optical axis (see Figure 6.2).

We used the following color scheme to support the three-step alignment approach: red

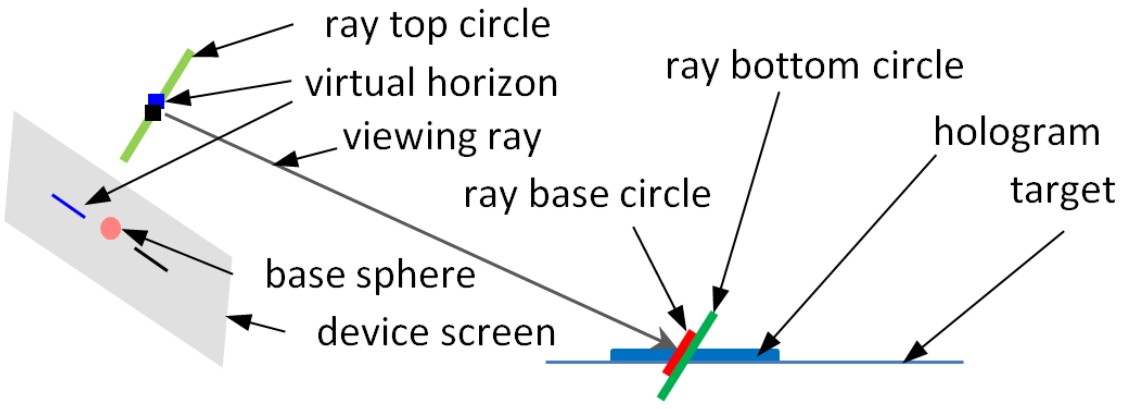

Figure 6.1: Geometry of the proposed alignment approach. Matching the current view with a given reference view takes place by aligning the viewing ray direction, position (base sphere on the device screen with the ray base circle, ray top circle with the ray bottom circle) and orientation (virtual horizon on top of ray with the virtual horizon on the device screen).

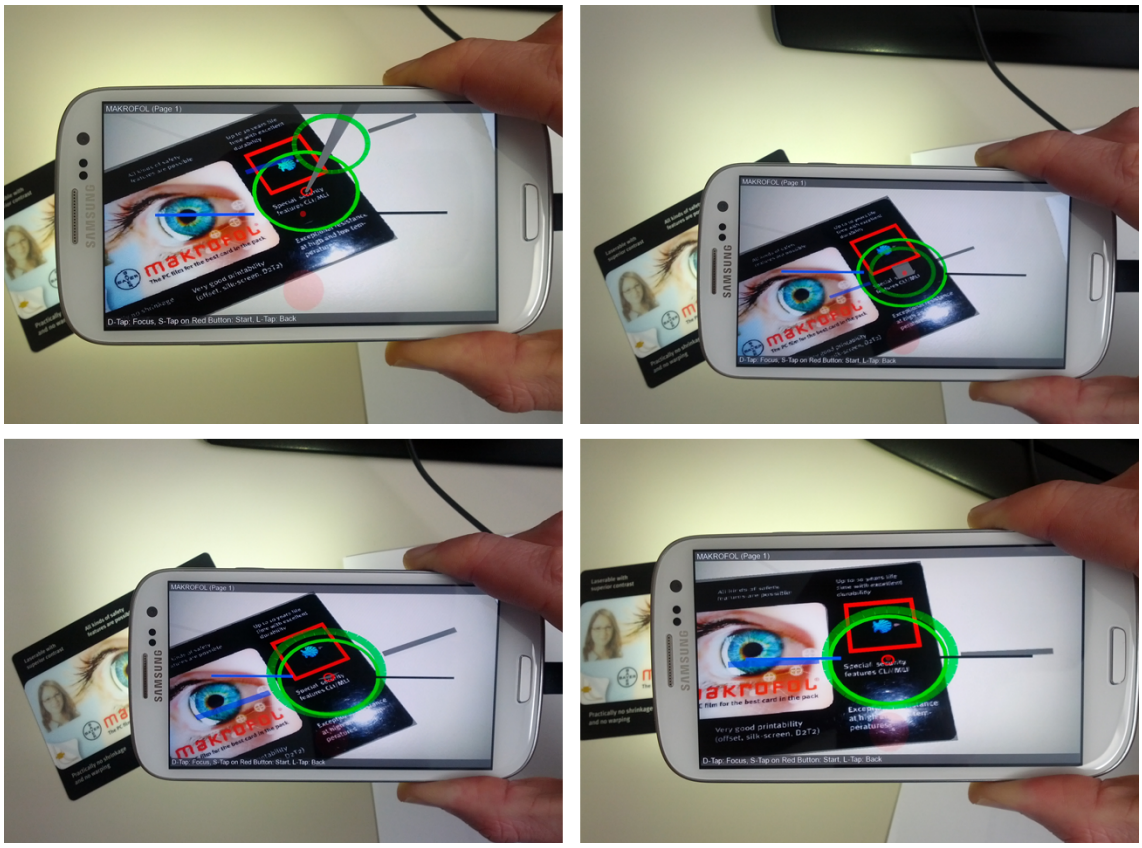

Figure 6.2: Exemplary alignment sequence: Not aligned (top left). Aligning direction using iron sights (top right). Adjusting distance (bottom left). Aligning rotation using the virtual horizon (bottom right).

for the on-screen sphere, small ray base circle, green for the big ray base circle, top ray circle and blue/yellow for the virtual horizon. Depending on the most similar (w.r.t. orientation) reference pose, an automatic pre-selection is carried out by the system, drawing the full iron sights and virtual horizon setup for the selected reference pose only. Whenever the user makes a selection, the color of the reference ray is adapted. The user gets a short summary of her decisions, when viewing the setup from farther away, and knows where no decision was recorded up to that point. We draw the last captured ray associated with the current reference pose, so that the user can get an impression of how well the captured views fit (see Figure 6.3 and Figure 6.2).

Runtime Operation: During verification, image capturing is triggered by the user, when the alignment of a reference pose and the current pose is deemed close enough for accurate visual feedback. In this case, an auto-focus operation is triggered, and the tracking pose is checked for stability, before the current frame and corresponding pose are recorded. This is to avoid recording of pose jitter or blurry patches. We assume the

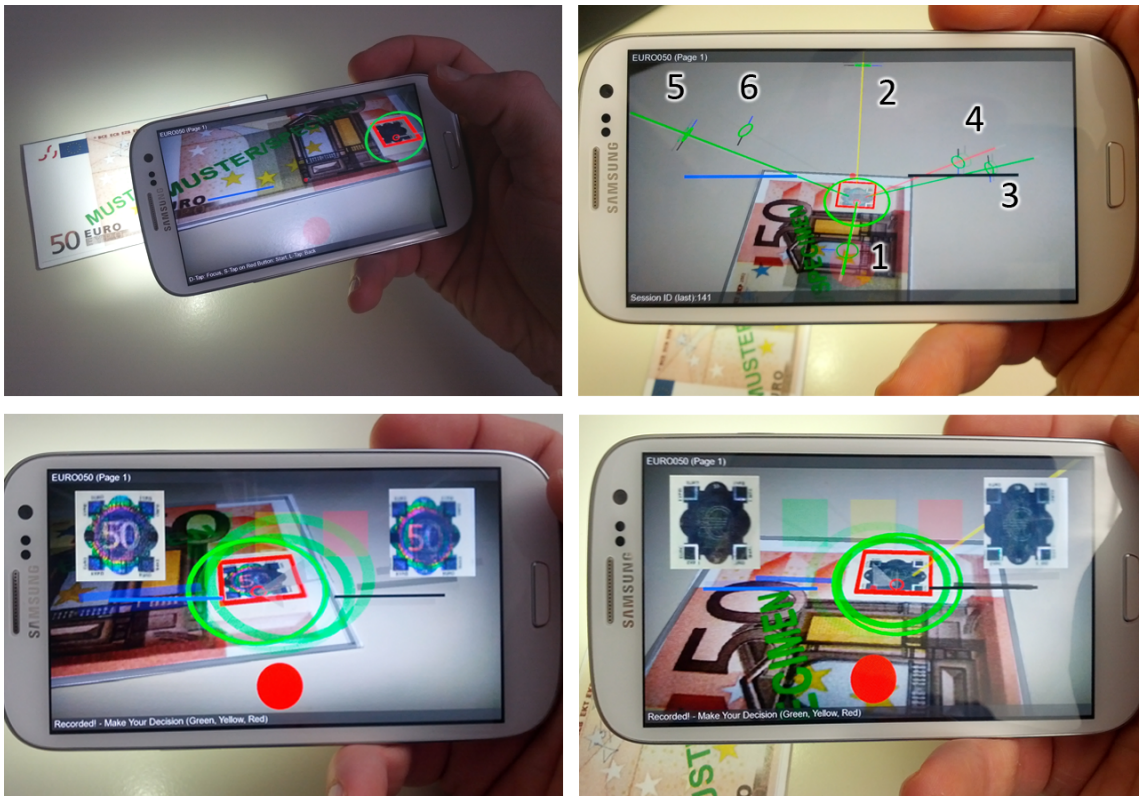

Figure 6.3: Our interactive system for verification of view-dependent elements performs SVBRDF capture using the built-in LED on the mobile device (top-left). The user gets an overview of relevant views for verification, which are color-coded w.r.t. the decision of the user (right, note the number attached to each view). The system allows the user to accurately match given reference views and to compare the changes of holographic or similar security elements with the corresponding reference appearances (bottom).

hologram to be planar and project the bounding box of the hologram into the image by using the current pose. We estimate an image transformation with respect to the hologram region on the undistorted template and subsequently warp the sub-image containing the hologram. Consequently, the appearance of the warped patch corresponds to the selected viewing direction. This allows for an efficient comparison. We display this patch side-by-side with a reference patch. This similarity must be rated by the user to express consent, uncertainty or rejection.

6.2.3 Evaluation

To test the feasibility of the proposed approach for mobile hologram verification, we determined several performance parameters with users in a pilot study. This study had two aims:

- Record the performance of users in target acquisition

- Provide a first comparison to a simple paper based method

For system performance, we wanted to know how accurate users can acquire the necessary viewing directions, given our guidance system. Understanding the potential accuracy limits is important for determining minimal angles and distances between views for verification and learning what differences the system has to tolerate. Moreover, we wanted to see if users can correctly verify a hologram using the current system. It is not clear if the representation on the screen under real lighting conditions is comparable and looks similar enough to users.

An additional goal was to analyze the potential for automation of the process, which includes automatic capture and matching of hologram patches.

6.2.3.1 Study Design and Apparatus

We followed a within-subjects study design, recording view alignment and matching performance, but also comparing the effects of the AR interface and a digital manual (DM, providing visual step-by-step instructions, see Figure 6.4) on several aspects in a hologram verification task. We investigated both performance based measures (alignment error, task completion time, error rates in matching and for the main task), and user experience (UX) dimensions (instrumental dimensions like usability, non-instrumental dimensions like hedonic stimulation and identity, and emotional dimensions like intrinsic motivation).

The experiment took place under controlled laboratory conditions. Specifically, the lighting was fixed to allow for comparable results in the digital manual condition. Both interfaces were deployed on Samsung Galaxy S3 smartphones. The tasks were carried out, while seated at a round table, but users were free to move around at any time (see Figure 6.5).

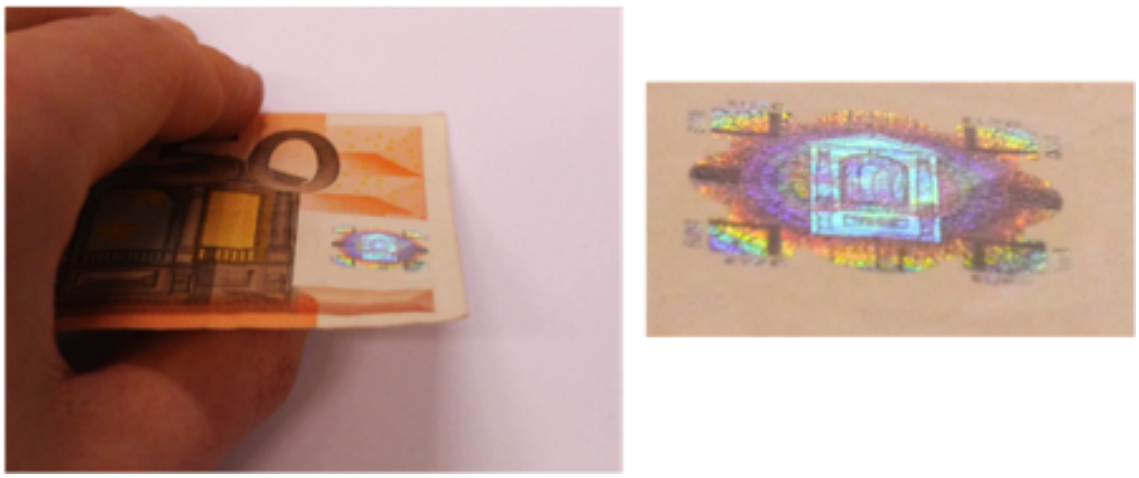

Figure 6.4: Exemplary view used in the digital manual. Overall image indicating the viewpoint (left). Zoomed image of the hologram patch (right).

Figure 6.5: Image showing table setup used during the study (left). Specimen banknote with window showing hologram to be checked by participants of the study (right).

6.2.3.2 Task and Procedure

As main task, we chose the verification of the hologram present on a 50 Euro banknote, which is one of the most often counterfeited banknotes in the Euro zone [38]. It must be noted that the holograms on banknotes with higher values (100, 200, 500 Euro) behave in a very similar way. The participants should inspect four holograms with each interface. Specifically, they were asked to view the hologram from six different viewpoints (depicting three different pictures: the banknote value, a window, and a doorway - see Figure 6.3 for view locations), but were free to stop the hologram verification, before completing all views if they already came up with a decision. They were instructed to compare the reference close-up view of the hologram with the view of the hologram that they were inspecting and decide if they were similar. We pointed out that the holograms do not have to match on a pixel-by-pixel view, but did not give any further hints on what similar meant, leaving this decision up to the participants. After inspecting the hologram from all six views, participants should come up with an overall decision on whether the hologram was a real one or a counterfeited one. We did not tell the participants at any time before, during or after the experiment if counterfeited (or real) holograms were among the ones they inspected. We used eight printed specimen notes in total (four per interface) and only left a hole for showing the underlying hologram of a real banknote (see Figure 6.5), to avoid that the checking of further security features of the banknote could influence the participants judgments. All employed holograms were real (no counterfeited hologram was used).

At the beginning of the experiment, users filled in a background questionnaire. They proceeded with a learning phase of the starting interface (AR or digital manual, counterbalanced) inspecting a hologram not related to banknotes followed by the main task.

After inspecting each banknote, participants briefly indicated their confidence in following aspects in an online questionnaire: Is the current hologram real or fake? Did the depicted reference viewpoints match the ones of the participants? Did the depicted reference close-up views match the ones the participants saw?

After checking the holograms on all four banknotes, participants completed intermediate questionnaires regarding workload and UX qualities of the interaction. They repeated the procedure (training, main task, questionnaires) with the second interface. At the end of the study, a short semi-structured interview was conducted, focusing on aspects observed during the participants' interactions with the interfaces. The overall duration of the experiment was around 60 minutes.

6.2.3.3 Participants

We conducted the study with 17 volunteers (1 female). Most participants reported to have considerable experience with computers and a high interest in technical matters. Only two volunteers reported not to own a smartphone or tablet. However, the majority (13 participants) had never checked a hologram before. Three of the participants were English speaking, but all instructions and questionnaires were given to the participants in either German or English.

6.2.3.4 Data collection

Within the experiment, we collected device, video and survey data complemented with photos and notes. For the AR system, we recorded camera poses and user interactions and captured hologram patch data along with task completion time. In case of the digital manual (DM), we measured the task completion time with a separate clock. In addition, the actions of the users were video-taped. Besides quantitative analysis of data, we employed several subjective scales to capture both general UX dimensions as well as task-specific aspects. Specifically, we employed the Nasa TLX for workload assessment [58], AttrakDiff [67] for capturing hedonic (stimulation, identity) and pragmatic UX dimensions and the interest/enjoyment and value/usefulness sub-scales of the Intrinsic Motivation Inventory (IMI) [109]. We analyzed quantitative data with the R statistical package and Microsoft Excel. Null hypothesis significance testing (NHST) was carried out at the 0.05 level. For the positional and orientation data, we treated all data outside the 2.5% and 97.5% percentiles as outliers. The percentiles were computed on the aggregated data over all views.

6.2.3.5 Results

We first analyze user performance in view navigation by comparison with the six given reference views at all relevant events. The subsequent analysis of patch similarity using image-based measures gives an impression on the performance of the proposed approach for

mobile SVBRDF capture. Then, we provide results on task-level performance (hologram verification) for the AR system and the DM, which attributes to patch similarity rated by the user and the ability to come up with a final decision. Finally we provide results related to the user's subjective assessment.

One participant took significantly longer for the proposed tasks than was suggested. As this behavior was limited to a single person, we consider the associated runs to be outliers and do not use the associated data in the evaluation.

Maneuvering to Target Poses: We analyzed data corresponding to all selected views during the study. Ranges of alignment errors in translation and rotation give a hint on the level of accuracy attainable with our guidance approach (see Figure 6.6). The range of translation error is -8mm to 10mm. The range of rotation error is -8 to 8 degrees. Overall, the largest error is encountered with view number 4. This was the first view typically selected by most of the participants, when they were still gaining familiarity with the system.

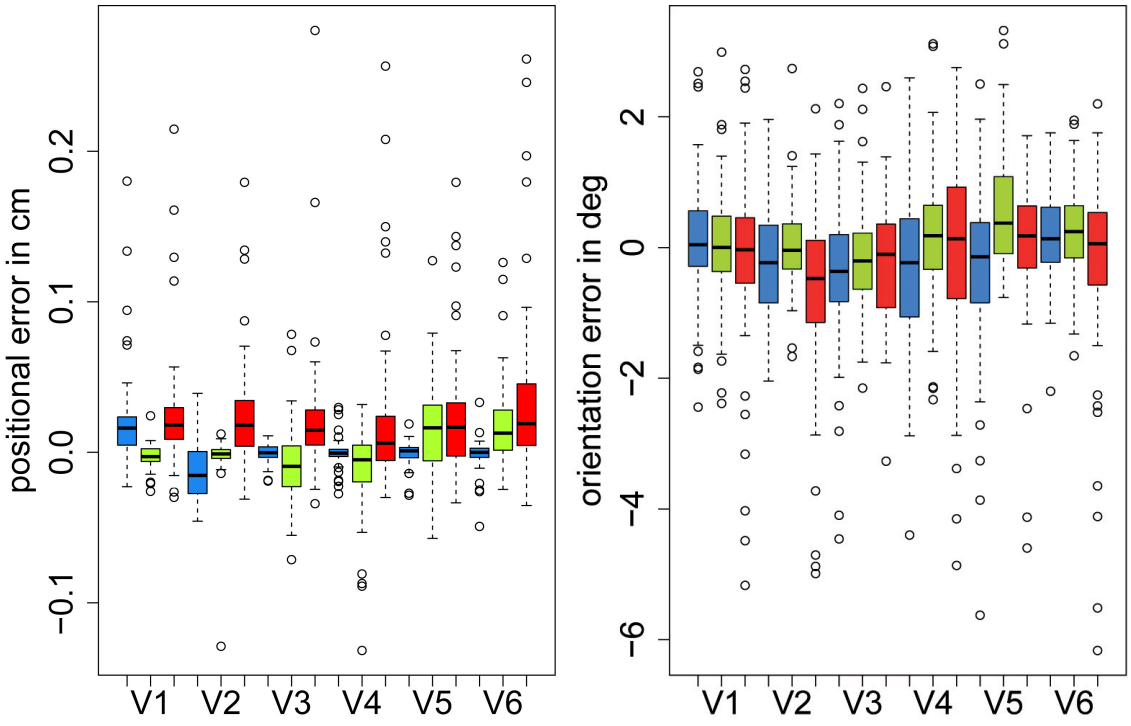

Figure 6.6: Alignment errors for different views of the hologram captured in the user study. Translation (left). Rotation (right). Axis color-coded: x...red, y...green, z...blue

Another way to assess the performance of using the guidance system is to compare the captured patches with a suitable image similarity metric. We register reference and captured patches using optical flow [134] and use NCC as our measure for patch similarity. The optical flow correction is to account for inaccuracies due to unstable tracking (see

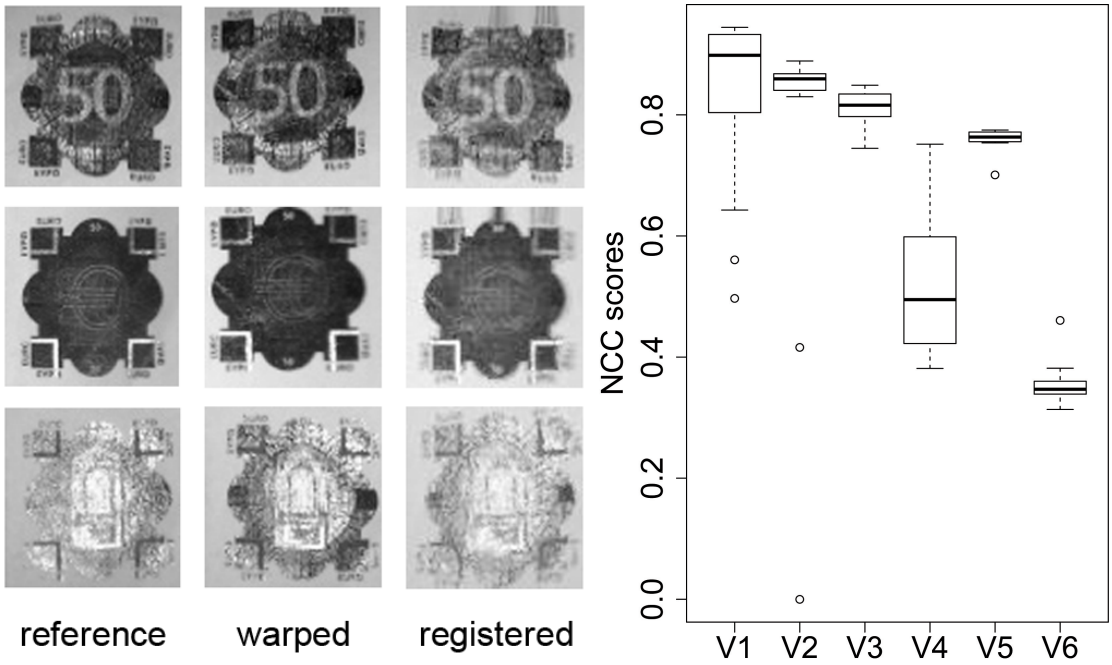

Figure 6.7: Matching registered patches: reference, warped image, registered image (left). NCC scores with registered images for different views (right).

Figure 6.7). In this setup, four out of the six views obtain average NCC scores above 0.75. This suggests that the proposed setup for SVBRDF capture allows acquisition of hologram patches for non-expert users, despite varying lighting conditions and limits in pose accuracy. Two views have very low NCC scores, however. Again, one of them is the view most users approached first, when they cannot be considered entirely familiar with the system.

Task Performance: Regarding the task completion time, the medians of the AR and the DM interface were 188 and 103 seconds, respectively (see Figure 6.8, left). As the data was not normal distributed, a two-tailed Wilcoxon signed-rank test was employed and showed that there is a significant effect of interface ($W = 1687, Z = 4.48, p < 0.05, r = 0.48$) on task completion time.

Participants rated how sure they were that the banknotes are real and fake for each banknote (see Figure 6.8, middle). In addition, they rated how confident they were that the individual hologram views corresponded to the reference close up views and how confident they were that their viewpoints corresponded to the reference viewpoints (camera poses). For the pooled results (over all four banknotes), a two-tailed Wilcoxon signed-rank test showed no significant effect of interface on any of those ratings.

Subjective Assessment: We used the NASA TLX weighted scores scheme to assess subjective demands. For computation of the scores, we used both the magnitude of load

(ratings) and sources of load (weights), which evaluate the contribution of each factor. The ratings for demands on subject and for task interaction are shown in Figure 6.8. Two-tailed Wilcoxon signed-rank tests indicated a significant effect of interface ($W = 98, Z = 2.13, p < 0.05, r = 0.37$) on physical demand (MD for AR: 14.67, MD for DM: 3.33) and a significant effect of interface ($W = 111, Z = 2.32, p < 0.05, r = 0.40$) on temporal demand (MD for AR: 5.67, MD for DM: 4.00). There were no significant differences in NASA TLX weighted scores for the other dimensions.

The AttrakDiff questionnaire is an instrument for measuring the attractiveness of an interactive system along pragmatic and hedonic user experience qualities. Paired two-tailed t-tests were conducted to compare the effects of the interfaces on the pragmatic quality (PQ), hedonic quality identity (HQ-I), and hedonic quality stimulation (HQ-S). Each subscale consists of seven items with a bipolar rating scale. We used five-item scales and averaged the ratings of all seven items for each subscale. Group differences for UX qualities PQ, HQ-I and HQ-S between the AR and DM interface condition are reported in Table 6.1 and Figure 6.9. The interface had a significant effect on all dimensions, with the AR interface leading to a significant lower score for the pragmatic (usability) dimension (with a medium effect size), but significantly higher scores for the hedonic dimensions (with large effect sizes).

We also assessed the participant's intrinsic motivation through the IMI [109]. Specif-

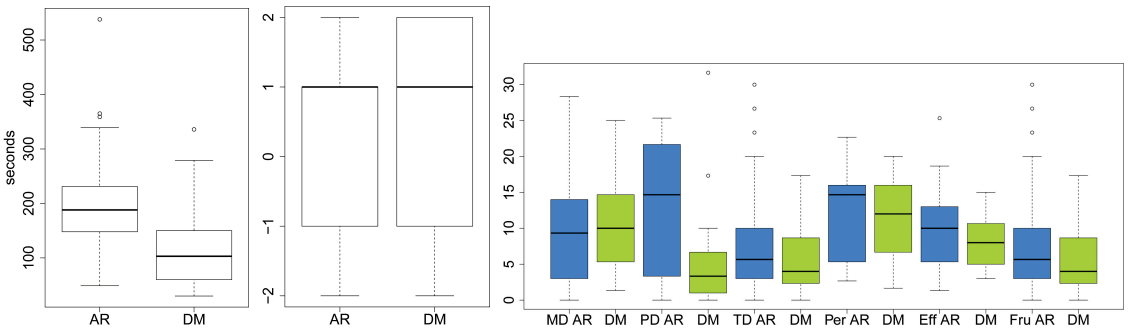

Figure 6.8: Task completion times for the augmented reality and digital manual interfaces (left) and agreement to 'I think the hologram is real' (middle). Weighted NASA TLX dimensions for demands imposed on subject and for task interaction; MD: Mental Demand, PD: Physical Demand, TD: Temporal Demand, per: Performance, Eff: Effort, Fru: Frustration (right).

	AR		**DM**				
Quality	M	SD	M	SD	t(13)	p	Cohens's d
PQ	-.08	.37	.28	.37	-2.58	.02	.37
HQ-I	.42	.37	-.15	.60	3.20	.005	.78
HQ-S	.6	.39	-.54	.67	7.58	6e-7	1.92

Table 6.1: Group differences for UX qualities PQ, HQ-I and HQ-S between the AR and DM interface condition.

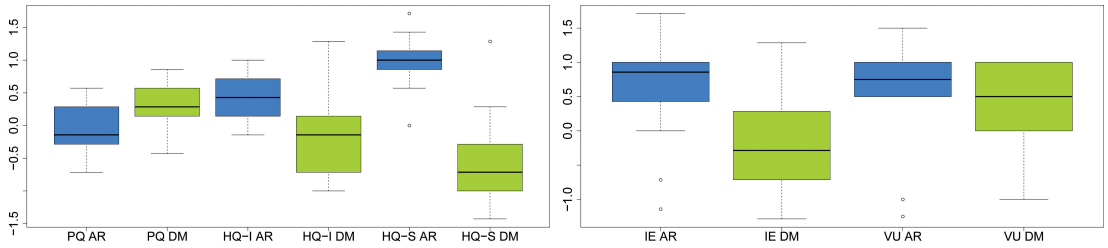

Figure 6.9: Left: AttrakDiff scores for Pragmatic Quality (PQ), Hedonic Identity (HQ-I), and Hedonic Stimulation (HQ-S) on a 5-item bipolar scale. Right: IMI scores for Interest/Enjoyment (IE) and Value/Usefulness (VU).

ically, we employed the interest/enjoyment and value/usefulness subscales (5-point Likert scale). A two-tailed Wilcoxon signed-rank test indicated significant effect for AR ($MD : 0.86$) and DM ($MD : -0.29$) on Interest/Enjoyment ($W = 123, p < .05, r = .38$). There was no effect on value/usefulness (see also Figure 6.9).

6.2.4 Discussion

The results obtained with the proposed approaches for hologram capture and user navigation demonstrate that non-experts regarding document security can record different appearances of hologram patches with consumer hardware.

More specifically, users were able to reach the six views used in the study with reasonable accuracy (maximum range of translation error from -8 to 10 mm, maximum range of rotation error from -8 to 8 degrees; see Figure 6.6). It must be noted that the used specimen banknote did not remain entirely planar during the study. Although potentially leading to larger errors, real banknotes often suffer from similar deformations. Consequently, several users commented that final alignment was tedious and should be automated.

Patch similarity computed after registration gave NCC scores greater than 0.75 for four of the six views (see Figure 6.7). While the pixel-wise registration improved NCC scores noticeably, the obtained pose accuracy was close enough to the reference view, so that the appearance of the view-dependent elements was correct. Thus, while we need to automatically correct for small pose variances, the correct sector for the view-dependent appearance was usually selected.

On the task level, the AR system shows similar verification performance to a digital manual, but longer task completion times and higher physical and temporal demand (see Figure 6.8). This is reasonable, because users are forced to move to the right pose, which ensures repeatable conditions and reasonable matching of patches. However, none of the evaluated interfaces was able to provide clear evidence whether a hologram is real or not. While the AR system was also not rated better in terms of usability (pragmatic quality), both the AttraktDiff and IMI (see Figure 6.9) questionnaires indicated significant higher ratings for hedonic dimensions and interest/enjoyment. This could indicate a higher motivational value for non-professional users to employ the AR system for verification.

Figure 6.10: Improved user interface (left): Top-left patch showing reference data for the nearest reference view, top-right patch showing captured data by the user. Bottom-left patch showing live view of the warped hologram patch. Additional elements represent visual ranges for easier alignment. Automatic recapture (right): The hologram is recaptured when a more suitable pose is encountered.

While most users requested automatic capture and patch matching, some users also suggested a summary page or the possibility to have a live view of the warped hologram patch. We redesigned the user interface to address these issues and to support a more natural workflow for verification. We incorporated a real-time view of the warped patch and added automatic recapture of the hologram, whenever a better match concerning pose is encountered (see Figure 6.10). We also provide cues for each of the steps required during alignment in the form of additional graphical elements representing alignment ranges. Finally, we show the reference patch and the best recorded patch concerning pose for the nearest reference frame. The user can now continuously inspect the hologram, get instant visual feedback and modify local decisions on validity.

We conducted an informal study with seven of the original participants, comparing the updated user interface with the previous iteration. Five of seven users felt more confident (two equal) on their decision concerning validity. However, six of seven users rated temporal effort to be equal (one less). Users verbally reported that they found the live-view and the alignment ranges to be useful. Regarding cognitive and physical strain, five of seven users rated the system to be equal to the previous iteration (two less straining). Two users asked for automatic image capture to avoid camera shake.

Not being able to work with actual fakes in the study certainly limits insights concerning practical usability. However, credible fake documents and in particular holograms are difficult to produce or acquire. The challenge is to get hold of samples which are not immediately identified as fake but strongly resemble genuine items. This also means that simple photocopies are of limited value. However, the approach could be evaluated with holograms from different documents embedded in a generic looking surround or even with rotated or possibly thermally treated holograms. The latter would allow to gain more insights w.r.t. practical usability.

All in all these can be considered encouraging results for building both manual and automatic mobile hologram verification systems.

6.3 Efficient User Interfaces

We aim to improve upon the efficiency of mobile hologram verification by proposing special task-oriented user interfaces. They avoid manual interaction, such as tapping on the screen, for reasons of accuracy and efficiency. Instead, image capture is triggered automatically, when the user is in a suitable position and matching with reference information is carried out in the background. With these setups, an automatic decision is available immediately after image capture.

We observed that many holograms feature similar appearances in very different locations. Consequently, one could think of rejecting the entire pose information during matching and just taking care that the user is pointing towards the hologram. However, this does not seem feasible, since users cannot be expected to sample the entire hemisphere thoroughly without guidance. It is mandatory to consider the viewing direction in order to get a good coverage of the pose space and provide a reliable exit mechanism. Using information about the viewing direction, when matching, provides additional security.

An obvious approach is to guide the user to align the mobile device with exactly those view points, which are associated with the selected reference data. Alternatively, a portion of space can be visualized for sampling by the user, which requires coverage of a larger region instead of given positions. Combining both approaches leads to a hybrid variant, which uses a comparatively small region for sampling relevant data. In the following, we cover the design of these approaches in more detail. In favor of usability, we decided to omit an explicit check of the in-plane rotation of reference views during matching. This is motivated by the fact that when views are placed on a hemisphere, reasonable results can be achieved by just rotating the target.

6.3.1 Alignment Interface

Sampling holograms can be treated as an alignment task, where users have to point at the center of the element, align with the viewing direction using iron sights, match the rotation along the viewing ray using a virtual horizon and take care of the recording distance [62]. Although this causes a lot of strain, we believe that a careful design in conjunction with automatic recording and matching can lead to considerable gains in efficiency. This could make the alignment approach a strong competitor for constrained approaches.

We propose an improved alignment interface, which was designed in an iterative process involving continuous user feedback. A sketch of the elements involved can be seen in Figure 6.11. We observed that users often had trouble matching the overall orientation of the document and the device with the original approach. Not being able to do so makes the overall alignment process more tedious. Consequently, the revised approach starts with coarse alignment of document-device orientation. We project the camera center and the top point of a reference pose down on the target surface and compute the relative angle as a rough indicator of initial alignment. This can be visualized as a color-coded

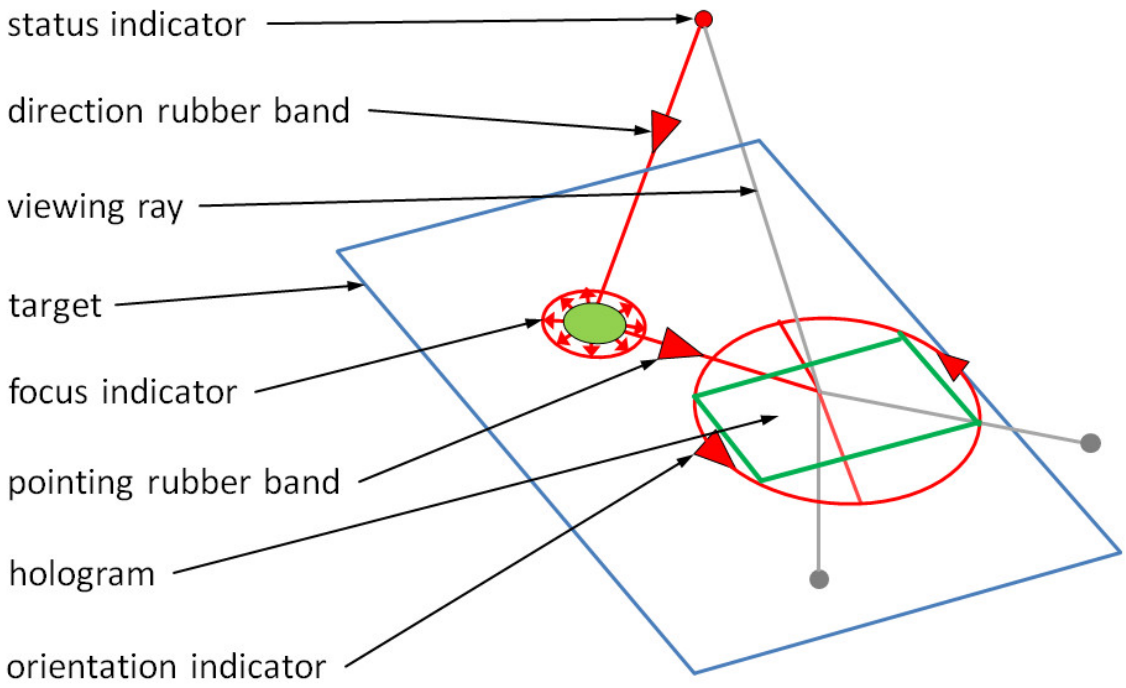

Figure 6.11: Geometry of the revised alignment approach. Matching takes place by alignment of target rotation and pointing with the indicator at the element. Finally, the viewing direction is refined using the direction rubber band at an acceptable viewing distance.

indicator within a circle around the element. Depending on the sign of the computed error, arrows are placed on the circle to indicate the required movement of the target. Upon successful alignment (within a certain range), we proceed with more accurate indicators for the viewing direction. We use animated rubber bands as indicators for pointing at the element, but also for the vertical angle on the hemisphere. In both cases, the goal is to follow the animated arrows in order to shrink down the rubber band into a point (see Figure 6.12). Finally, a focus indicator is realized as a scaled sphere placed at the base point of the current viewing direction on the target. Animated, directed arrows indicate the required direction of movement. Note that we perform an initial focus operation at the first view to be aligned and keep this setting throughout the process.

Views are captured sequentially, with feedback on the overall progress of the operation. This aims to reduce visual clutter for the user. Upon successful alignment, several frames are recorded from the live-video stream and automatically matched against prerecorded reference data. From these measurements, the one having the highest matching score is selected as the result patch for the user. During the process, we provide guidance towards the desired direction, but also feedback regarding the quality of alignment. Similar to the previous approaches, we aim to minimize the required movements for the user by automatic selection of the nearest view. A live-view of the rectified hologram patch is constantly displayed during spatial interaction in order to provide visual feedback of the

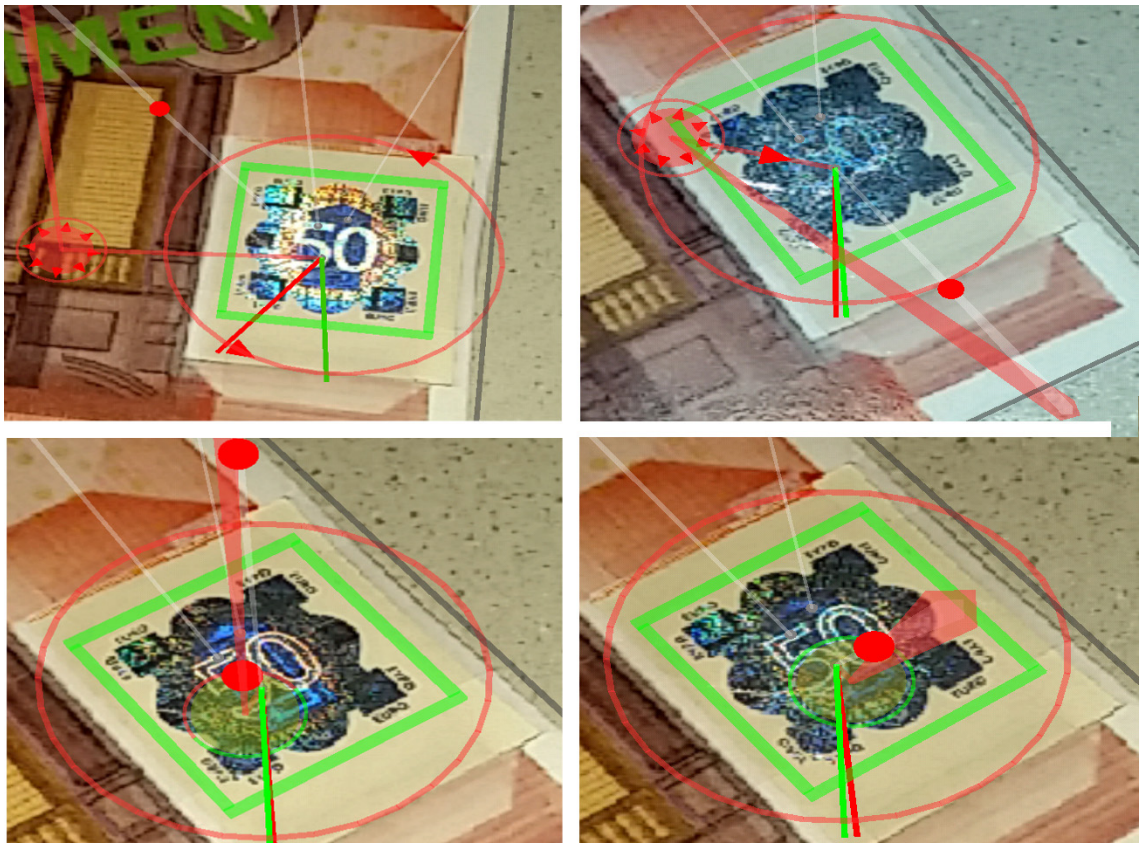

Figure 6.12: Exemplary alignment sequence: Not aligned (top left). Aligning target rotation (top right). Pointing at target (bottom left). Aligning viewing direction along hemisphere arc (bottom right).

changes in appearance with varying recording position (see Figure 6.12 for an exemplary alignment sequence).

After recording each of the views, a summary including the current overall decision (genuine/fake) is presented to the user (see Figure 6.16). The user may skim through the captured views and compare them side-by-side with the expected reference data. If the system suggestion is revised by the user, an overall similarity score is recomputed, which eventually changes the final decision. The user may also re-record certain views in order to get a better basis for the final decision. This can be done in the summary for the current view and works for all the approaches described here.

6.3.2 Constrained Navigation Interface

The task can also be treated within a constrained navigation framework. The idea is to guide the user to sample larger portions of space instead of aligning with single views. By giving more freedom to the user, this can reduce workload and task completion time.

The initial step is to guide the user to point at the hologram as required by the

recording setup. We provide guidance using an animated rubber band, which shows a moving arrow, once outside a given radius away from the element (see Figures 6.13, 6.14). The capture distance needs to be adjusted as a starting point for an auto-focus operation, so that the assumption about the flashlight being the dominant light-source holds. For this purpose, we scale the entire widget and require the user to adjust the distance, so that the outer ring of the widget stays within the given distance bounds.

Although the robot recording operates on a hemisphere, it does not seem reasonable to apply this concept directly. An augmented hemisphere would certainly lead to coverage of the entire space, but not necessarily in the shortest possible time. With an augmented hemisphere, the most obvious movement is to scan hull slices and then rotate the document for the next slice. We empirically verified that changing orientation from an orthogonal starting point (conic) is much faster than target rotation with slice-scanning.

In favor of efficiently treating both originals and fakes, the user should be guided towards different viewing directions or ranges. We propose a 2D orientation map (projection of the conic space) [69] for this task. It is divided into slices that are aligned on one or more tracks. The current position on the map is visualized by a cursor, and the current slice is also highlighted. The cursor position is corrected by the target orientation, so that the movement direction always corresponds to the orientation of the device (see Figures 6.13, 6.14). In general, it is not sufficient to just capture a single shot inside each slice. We record several shots per slice, that differ at least by a given angular threshold. The exact amount is automatically calculated, taking into account the area of the slice. Consequently, the user can move freely inside the pie slices during the process. The tiny

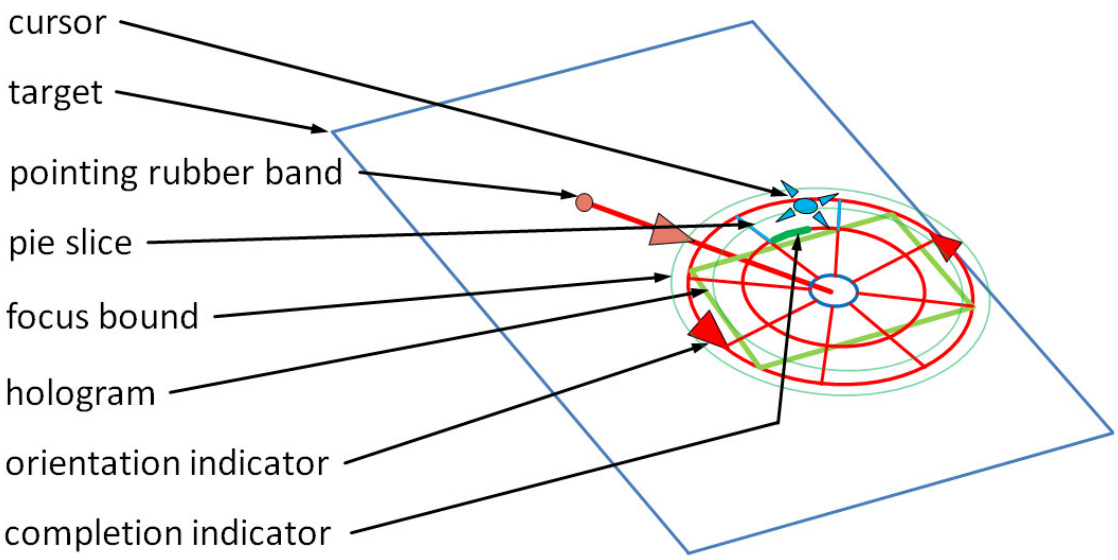

Figure 6.13: Geometry of the proposed constrained navigation approach for sampling the hologram. The user is guided to point at the element, and a cursor is controlled by the 2D orientation on an augmented pie, divided into slices and tracks.

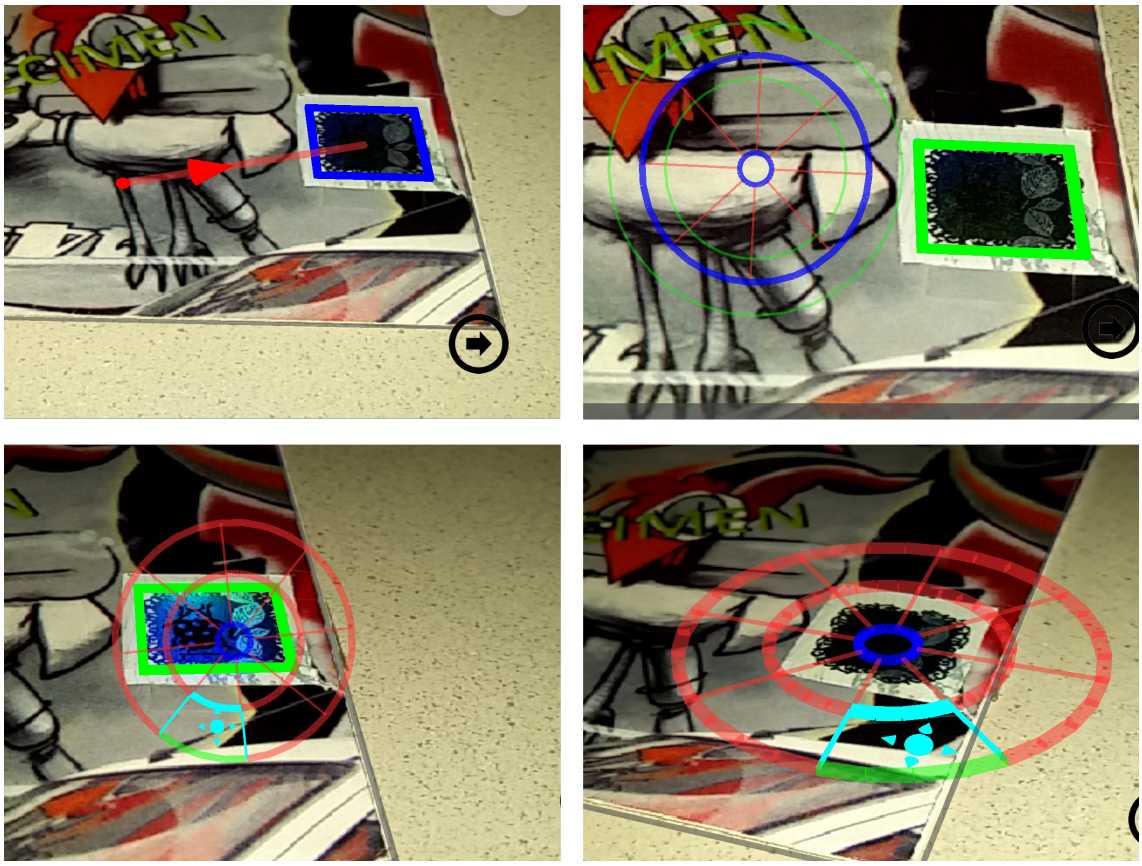

Figure 6.14: We guide the user to point at the element using an animated rubber band (top-left). Focus adjustment showing the layout of the orientation map and green distance bounds (top-right). Constrained navigation UI with pie slices (bottom-left). Augmentation directly onto the document/element (bottom-right).

arrows around the cursor serve as movement indicators. Whenever the user remains static inside a non-completed slice, flashing arrows remind to move on. The upper arc defined by a (sub-)slice is used as a completion indicator, which switches from red to green with increasing slice coverage. The orientation map is realized as a widget placed in the screen plane (2D-CON) or augmented onto the target (AR-CON).

In a pilot study, we tried using either no visual information on the capturing procedure or a progress bar without any orientation information. Using no visualization at all gave the best completion time, but also the lowest spatial coverage. In the following, we dropped the interface without guidance and the progress bar. It must be noted that even with the AR-CON interface, not all participants sampled the entire hologram. Consequently, we went to incorporate slightly more guidance with the goal to only check pie slices containing a reference view (see Figure 6.15).

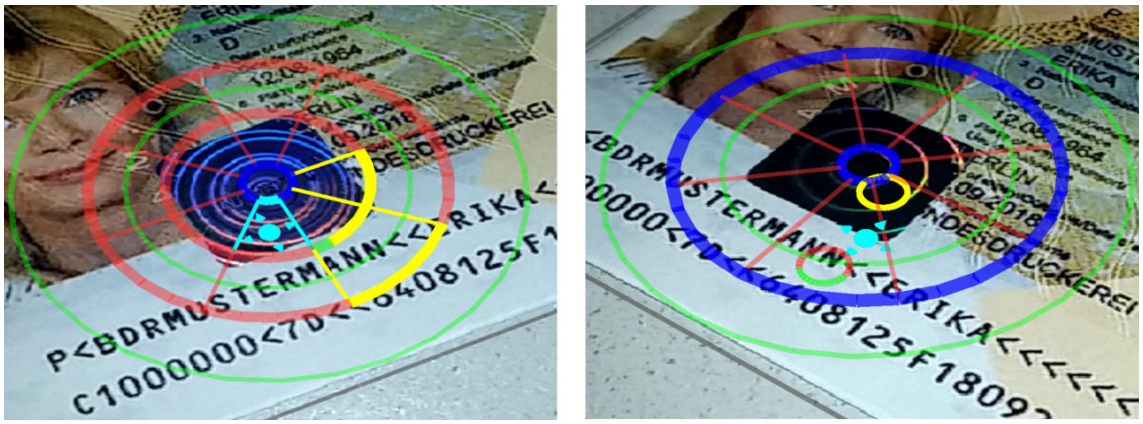

Figure 6.15: AR UIs with guidance for interesting subspaces. Either pie-slices (AR-CON, left) or circular regions (AR-HYB, right) are indicated for sampling by the user.

6.3.3 Hybrid Interface

The location of reference views cannot be mapped straightforward to pie slices. It may be necessary to associate several pie slices with a single reference view, increasing the amount of slices to be checked. Since the number is generally much lower than the total number of pie slices, we use small regions on the augmented map around reference locations, which also serve as local completion indicators (AR-HYB, Figure 6.15).

These two UIs were evaluated in another pre-study, this time involving a demonstration phase. According to the results obtained, AR-HYB had a much lower task completion time compared with AR-CON. Users were able to complete the task using both approaches (perfect coverage of interesting slices/regions) and obtained reasonable patch-matching scores. Users generally gave very positive ratings concerning the type of guidance and overall usefulness of the application, with a clear preference for AR-HYB. Motivated by user demand and our own reasoning, this clearly moved the approach more in the direction of an alignment task. As we consider our informal studies only suitable for guiding the design process, we conducted a more detailed evaluation.

6.3.4 Evaluation

We evaluated the most promising candidate for constrained navigation (CON) and the hybrid approach (HYB, see Figure 6.15) against the alignment UI (ALI, see Figure 6.12). After image capture, a summary is presented to the user (see Figure 6.16) independent of the UI used for capture. The global system decision is communicated via a colored square (green...valid, yellow...unsure, red...invalid) to the user. Each reference has its own page, showing the reference data on the left side of the screen and the best recorded match on the right side, along with a local rating by the system, which can be changed by the user in case of doubt. It must be noted that we also monitored distance as capture condition, so that the users had to stay within the allowed distance range for the CON and HYB

Figure 6.16: User interfaces for hologram verification: Constrained navigation (top-left), alignment (top-right) and hybrid user interfaces (bottom-left) are designed, implemented and evaluated within a user study. They allow to reliably capture image data suitable for automatic verification. Results are presented to the user in a summary (bottom-right).

interfaces. We manually selected two reference views per hologram with a visually equal spatial distribution.

6.3.4.1 Study Design and Tasks

According to a domain expert we consulted, professionals can identify most fake documents or holograms within a few seconds. The focus of the following study is on laypersons without advanced domain knowledge or experience, using an off-the-shelf smartphone for hologram inspection.

We designed a within-subjects study to compare both the performance and user experience aspects of the three aforementioned user interfaces for hologram verification.

The study had two independent factors: interface and hologram. The independent variable of main interest was interface (with three levels: ALI, CON, HYB). We modeled hologram as fixed effect (four level), since the holograms were deliberately selected (and not randomly sampled from a population) in order to represent intensity-dominated and shape-dominated samples including common mixtures.

For each of the four holograms, we selected the corresponding reference views with the goal of minimizing the variance an individual hologram could have on the results. Dependent variables of interest were task completion time (both capture and decision time), system performance (how well the system could verify the validity of the hologram), user performance (how well the user could verify the validity of the hologram), and user

experience measures (usability, workload, hedonic and motivational aspects).

For each interface, the actual verification procedure started upon pointing the center of the screen at the element and tapping on it. For the ALI interface, the user had to align the rotation of the document with the current reference view (azimuthal angle), point at the center of the hologram and adjust the viewing direction (polar angle) along with the capture distance. In case of the CON interface, the user had to point at the element, following the base rubber band. The orientation cursor had to be moved inside the indicated (connected) pie slices by changing the azimuth and inclination angles through device movement and monitoring the operating distance. The HYB interface had to be operated in a similar way. However, the cursor had to be aligned and moved inside small circular regions. Upon successful sampling, the system summary/system decision was presented to the user.

6.3.4.2 Apparatus and Data Collection

We conducted the study in a lab with illumination from the ceiling enabled (fluorescent lamps). In order to minimize variations induced by daylight changes, we kept the blinds of the room closed throughout the entire study.

All user interfaces were integrated into a single Android application running on the Samsung Galaxy S5 mobile phone (Android 4.4.2) and using the built-in camera with LED flashlight enabled. Reference data for verification was recorded with our robot using the same device (see Section 5.3.3.1).

We used four holograms as shown in Figure 6.17, each on a different base document. With our choice of samples and reference data, we aimed to address the non-trivial case of hologram substitution, since that is rather common according to a document expert we consulted for our study. Although some of the views we selected (i.e., black patches) may not resemble the typical appearance of holograms for the public, we believe that the large visual difference w.r.t. the other image in the pair justifies their use.

We collected data for evaluation through automatic logging on the test device itself, questionnaires and interviews. For data analysis, we used Matlab, R, and SPSS. Null hypothesis significance test were carried out at a .05 significance level, if not otherwise noted.

6.3.4.3 Procedure

Each participant was informed about the study purpose and the approximate length prior to the start of the study. The participants filled out a demographic questionnaire and then conducted the Vandenberg and Kuse mental rotation test [170]. They were informed that they would test a total of 12 holograms with three user interfaces (four holograms per interface). Although 12 holograms were shown to the participants as a stack, only a subset of four holograms was used for all interfaces (see Figure 6.17).

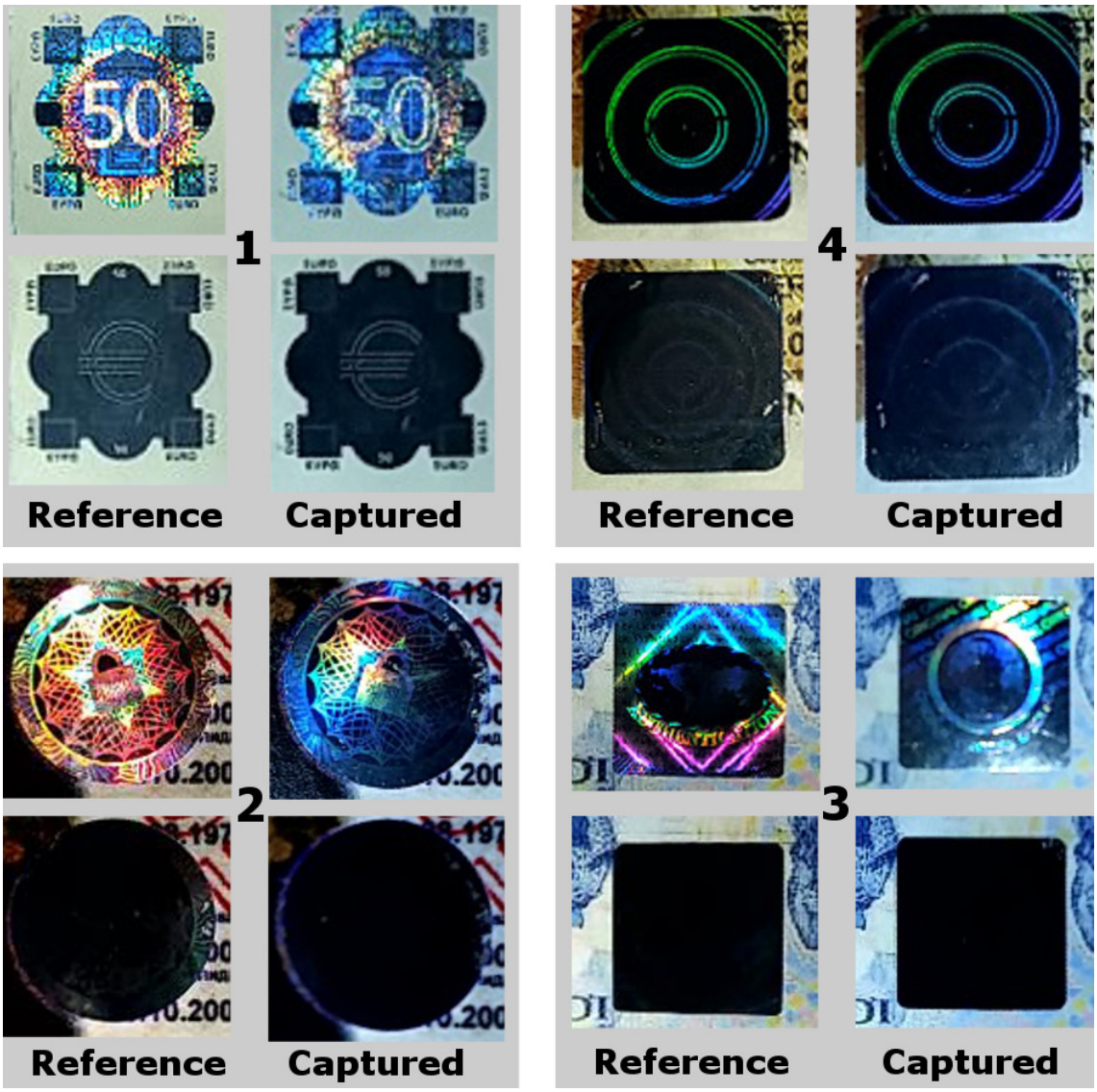

Figure 6.17: Samples used in our study. We evaluated all user interfaces with two original (no. 1, 4 - top row) and two fake (no. 2, 3 - bottom row) holograms, where each was placed on a different document template. Reference information recorded with the robot setup is used by the system for matching, while the other images are exemplary recordings during verification by the user.

The following procedure was repeated for all three user interfaces. A training phase with both a correct and a fake document (not appearing in the actual study) was conducted. This also included an explanation of application controls along with document classification and tracking. Participants could test the interface as long as they liked (on average less than five minutes). After feeling comfortable with the interface, participants were asked to use the current interface to capture four holograms, one at a time. After capturing a single hologram, the system presented its decision on the validity of the single

views and an overall decision (valid, unsure, invalid). After seeing the system decision, the participants were asked to fill out a post-task questionnaire, in which they were asked to assess the validity of the hologram on their own (5-item bipolar scale: I am totally sure that the hologram is fake ... neutral ... I am totally sure that the hologram is valid). After validating four holograms with the current interface, the users filled out a post-interface questionnaire (5-item Likert scale, ease-of-use and time items of the After Scenario Questionnaire [95]), the NASA TLX questionnaire (with weighting of items) [58], the AttrakDiff [67] and Intrinsic Motivation Inventory questionnaires [109].

After having conducted this procedure for all three interfaces, the participants filled out a final questionnaire, in which they should choose their preferred interface (overall preference, which interface was fastest to use, which interface was easiest to use). Finally, they were asked about the reasons for their choices. Participants received a voucher worth 10 EUR for their time.

The starting order of both interface and hologram was counterbalanced. The tasks where grouped by interface. While each participant was exposed to each hologram three times, we took care to make them believe it was a separate hologram (by showing a staple of several documents and hiding from them which document was drawn out of the staple). Each participant was exposed to individual interface-hologram combinations exactly once during the study. The whole procedure took on average 90 minutes. Participants could take a break anytime they wanted.

6.3.4.4 Participants

19 volunteers (2 female, age $M = 26.8, SD = 4.46$) participated in the study. All except one participant owned at least one smartphone or tablet, where the majority (16) had been using it for at least one year. In general, participants reported to be interested in technology. Thirteen participants had already used an AR application at least once. Seven participants had never attempted to verify a hologram before. In the mental rotation test, the majority of participants scored reasonably ($M = 0.8, SD = 0.14$). With 19 participants assessing four holograms with three interfaces, we obtained 228 samples.

6.3.4.5 Hypotheses

Based on our observation and the insights gained during pre-studies, we had the following hypotheses: *H1:* The hybrid UI will be the fastest among all interfaces. *H2:* The alignment UI will be the most accurate one, but slow. *H3:* The constrained navigation UI will be the easiest to use.

The hybrid interface combines desirable elements from alignment (accurate end position) and constrained navigation (marked interaction space). With a small number of reference views, checking should be very fast (*H1*). The revised alignment interface should assure the most accurate capture positions and, consequently, has the best prospects for accurate matching and verification (*H2*). This might come at the cost of increased capture

time. The constrained navigation approach gives most freedom to the user. The pie slice layout could be familiar to users, although accuracy w.r.t. single reference views might not be as good and, by design, a bigger space needs to be sampled (*H3*).

6.3.4.6 Findings

We performed an analysis of task completion time, user and system performance and user experience aspects for hologram verification.

Task Completion Time: For capture time (the time from start of the task until the presentation of system results), a two-way within-subjects analysis of variance (ANOVA) showed a significant main effect for interface, $F(2, 36) = 3.60$, $p = .038$, *partial* $\eta^2 = .17$ and a significant main effect of hologram, $F(3, 54) = 4.04$, $p = .012$, *partial* $\eta^2 = .18$. The interaction between interface and hologram was not significant.

Multiple pairwise post-hoc comparisons with Bonferroni correction for interface revealed that the mean score for capture time (in seconds) for the hybrid interface ($M = 37.22, SD = 38.20$) was significantly different compared to alignment ($M = 57.01, SD = 55.77$) ($t(75) = 3.44, p = .001$), but not compared to constrained navigation ($M = 44.43, SD = 20.70$). Also, there was no significant difference between constrained navigation and alignment.

Multiple pairwise post-hoc comparisons with Bonferroni correction for hologram revealed that the mean score for capture time (in seconds) for hologram 2 ($M = 39.61, SD = 29.39$) was significantly different compared to hologram 4 ($M = 55.19, SD = 53.51$), $t(56) = -3.23, p = .002$, but not compared to hologram 1 ($M = 45.58, SD = 40.97$) or hologram 3. There were no other significant differences between holograms. Furthermore, there where no learning effects for either interface or hologram, as indicated by a within-subjects ANOVA.

The decision time (the time spent in the summary screens) over all interfaces was on average 18.45 seconds ($SD = 15.32$). A two-way within-subjects ANOVA showed no significant main effect for interface, but for hologram $F(3, 54) = 3.233, p = .029,$ *partial* $\eta^2 = .152$. However, multiple pairwise post-hoc comparisons with Bonferroni correction for hologram did not indicate any significant pairwise differences (hologram 1 $M = 17.7, SD = 12.72$, hologram 2 $M = 23.7, SD = 19.54$, hologram 3 $M = 15.53, SD = 8.85$, hologram 4 $M = 16.98, SD = 17.54$). The interaction between interface and hologram was not significant.

To summarize, the capture time using the hybrid interface was significantly faster than the alignment interface and for hologram 2 compared to hologram 4. For decision time, no pairwise significant differences could be found. There were no learning effects for interface or hologram.

User and System Performance: Over all participants and holograms, 79.6% of the users' decisions were correct (treating both items 'I am totally sure that the hologram is [in]valid' and 'I am sure that the hologram is [in]valid' as correct answers). For 12.5% of the decisions, the users where unsure if the hologram was valid or fake. An investigation of the effects of the predictors interface and hologram on the dichotomous dependent variable 'correctness of user decision' using logistic regression was statistically not significant. Note that we had to exclude one participant from this sub-evaluation due to incomplete data.

73.1% of the system decision were correct. The system was unsure if the hologram is valid or fake in 11% of all cases. As for user decision, we used logistic regression to investigate the effects of interface and hologram on the dichotomous dependent variable 'correctness of system decision'. The logistic regression model was statistically significant $X^2(5) = 58.83, p < .0001$, explained 37.5% (Nagelkerke's R^2) of the variance in system decision and correctly classified 81.5% of the cases. The Wald criterion demonstrated that hologram made a significant contribution to prediction ($Wald\ X^2(3) = 20.80, p < .0001$), but interface did not. The system only made correct decisions in 50.0% for hologram 1 (neutral: 27.8%, hologram 2 correct: 100%, 3 correct: 94.4%, 3 neutral: 0.04%), 4 correct: 74.1%, 4 neutral: 13.0%).

To summarize, users were able to correctly validate (decide if the hologram is valid or false) in 80% of the cases, but the system only in 73%. Hologram was a significant predictor for system decision, with a validation performance for hologram 1 of only 50.0%.

User Experience: We investigated ease of use and satisfaction with task duration with the ASQ, cognitive load with the NASA TLX, and hedonic and motivational aspects with AttrakDiff and Intrinsic Motivation Inventory questionnaires, after each participant had finished using a single interface.

A one-way Friedmann ANOVA by ranks did not indicate a significant effect of interface on ease-of-use. Similarly, for satisfaction with task duration (over all four holograms per interface), there was no significant effect of interface. Note that we had to exclude one participant from this sub-evaluation due to missing data.

For cognitive workload, as measured by NASA TLX, one-way Friedmann ANOVAs by ranks did not indicate significant effects of interface on the subscales (mental demand, physical demand, temporal demand, performance, effort, frustration) or the overall measure. Due to space reasons and the non-significance of the omnibus tests, we will not report further statistics here.

Similar, for pragmatic quality (PQ), hedonic quality - identity (HQI) and hedonic quality - stimulation (HQS), as measured by AttrakDiff, and for value-usefulness and interest-enjoyment as measured by the Intrinsic Motivation Inventory, one-way Friedmann ANOVAs by ranks did not indicate significant effects of interface.

In the final questionnaire, 47% of the participants indicated that CON was easiest to use (ALI: 21%, HYB: 32%), 42% indicated that CON was fastest to use (ALI: 16 % HYB: 42%) and 47% favored CON overall (ALI: 26.5%, HYB: 26.5%).

In summary, the statistical analysis could not indicate significant effects of the interfaces on usability, workload, hedonic qualities or intrinsic motivation. Still, about half of the participants preferred CON overall and indicated that it was easiest to use.

6.3.5 Discussion

Our analysis did not fully confirm hypothesis H1. The hybrid interface was the fastest one, taking roughly 40 s for image capture, being significantly faster than the alignment interface (which took around one minute for verification). However, the hybrid interface was not significantly faster than the constrained navigation interface (ca. 45 s).

While this is a significant improvement over related work ([62], but using up to six views), this is still a long time span and probably not feasible for a quick check in a real-world situation. However, as most checked documents will be originals, an early exit for such samples could further decrease checking time. As decision time did not vary significantly between interfaces, they are all suited to recording data for verification.

Around 73% of the system decisions were correct, which may seem rather low. As there was no significant effect of any interface, hypothesis H2 does not hold in this regard. If we only neglect wrong decisions (i.e., combine positive and neutral decisions), the system performance would still be below the combined rate for user decisions (system: 84% correct vs. user: 92% correct). It seems that users either came up with their own (more invariant) similarity measure during the study, or they used additional appearance information gathered through the sampling process for their decisions, which was not available to our system (e.g., due to non-matching viewing direction). However, most of the neutral system decisions (around 63%) were caused by hologram 1 (50 EUR banknote, see Figure 6.17). This hologram shows rainbow colors, which is a very difficult case for our matching approach. Together with the rather conservative parametrization of our system (avoiding false positives) and the encouraging results of hologram 4 (around 90% combined rate), we speculate that the type of hologram has considerable influence on its verifiability with the proposed approach.

While the statistical analysis did not indicate significant effects of interface and user experience measures, we obtained a large number of comments in the post-hoc interviews throughout the study. The HYB UI, being the fastest one, was described four times as 'intuitive', 'good to use' or 'easy' (CON: 7, ALI: 3). However, four participants reported that the movements required were initially not clear (CON: 4, ALI: 5). With the CON UI, four users recognized the freedom in movement. For the slow ALI UI, three users expressed their interest in that UI ('interesting', 'cool idea', 'visually best'). One user stated that it was 'easy to spot, what to do, but difficult to accomplish'. Two users also gave positive comments about the usefulness of the summary.

For the CON and HYB interface, one user suggested to always display the pointing rubber band, even when the widget is perfectly at the screen center. For CON, one user suggested an additional completion indicator for pie slices involving the pie region itself

instead of the border. The same user also suggested to use additional indicators for viewing ray alignment in the ALI UI.

Despite being the fastest one (around 40 s), the hybrid user interface did not receive the same degree of user consent as the CON interface when taking into account the comments. Users explicitly criticized the final alignment stage involved. As a take away, it seems that the most efficient interface does not necessarily reflect the general preference of the user. Such awareness should be considered for real-world deployments of mobile AR user interfaces requiring fine-grained maneuvering.

6.4 User-Friendly Parametrization

Both the gap in matching performance between the system and the operator as well as the improved, but still impractical verification time deserve additional attention. We integrated SSIM for matching into the mobile prototype for hologram verification and propose an alternative distribution of reference views. In this case, more reference information can be used for matching, while still requiring only small movements by the operator. We evaluate this setup within a user study regarding accuracy and task completion time.

6.4.1 Distribution of Views

With the goal of further reducing temporal effort, the selection of reference views to be checked becomes increasingly important. A selection focusing solely on differences in appearance (as used in a printed or digital manual) could be disadvantageous for mobile applications, since a large range in orientation needs to be spanned by the user. In contrast, a reasonable spatial positioning of reference views could lead to savings in task completion time. With the experience gained in prior experiments, it seems reasonable that the layout of reference views conforms with typical movements of users when examining holograms (see Appendix B). In order to allow the inspection of a hologram regardless of whether the document is lying on a desk or held in hand, reference views should be placed in the lower vertical direction of the orientation space (see Figure 6.18). Due to the observed movement along a path, it also seems reasonable to use a sequence of patterns for verification instead of single spots. In this case, more data is available for matching, which could lead to more robust decisions by the system.

6.4.2 Evaluation

We took four pairs of originals and substitutes used in a previous experiment (see Figure 5.15) and selected an alternative layout of reference views (see Figure 6.19) for the Hybrid user interface. This prototype was then evaluated within a user study with the goal to evaluate the accuracy of decisions by this modified and re-parametrized system as well as the temporal effort concerning image capture and decisions.

6.4.2.1 Procedure

Participants were informed about the study purpose and length, followed by a short investigation of demographic data. Then, a training phase was started in order to make the participant familiar with the checking procedure using a fake and an original document. Afterward, four pairs of documents (original, substitute - see Figure 5.15) had to be checked using the proposed approach. We rotated the sequence of these documents with each participant. During this process, relevant data such as timestamps for various actions, matching scores and system/user decisions on validity were recorded. After each hologram, the users were questioned about their decision on validity. After all runs, they were asked to rate the process as a whole and to give comments regarding their experience. We used a Samsung Galaxy S5 smartphone throughout the study.

Altogether, 24 users participated in the study (2 female, age $M = 29.54, SD = 5.54$). All but one user reported to own a smartphone for at least one year. In general, they described their affinity to technology as high to very high. Half of the participants reported to never have examined a hologram before.

6.4.2.2 Findings and Discussion

Temporal Effort and Accuracy: Holograms can be assessed by the system immediately after image capture, which takes approximately 15 seconds ($M = 14.97, SD = 8.59$).

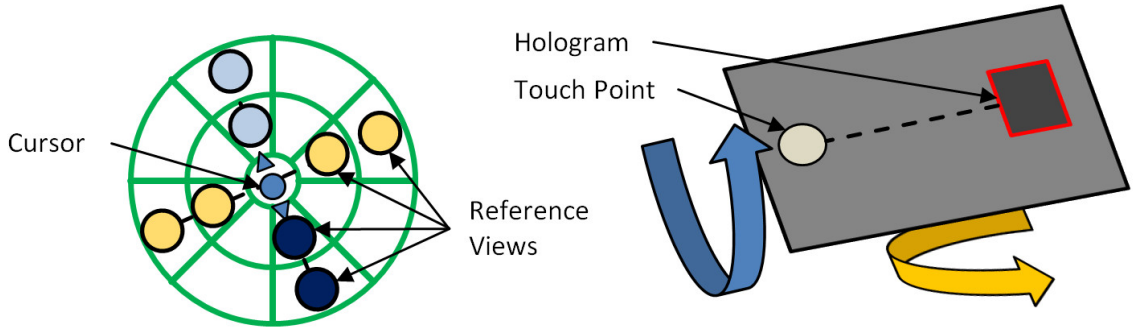

Figure 6.18: Alternative layout of reference views (orientation) for hologram verification on movement paths. Reference views should be placed on a vertical path, making the device face towards the user in case the document is lying on a surface (dark-blue circles).

Figure 6.19: Exemplary selection of reference views on the lower half of a vertical path.

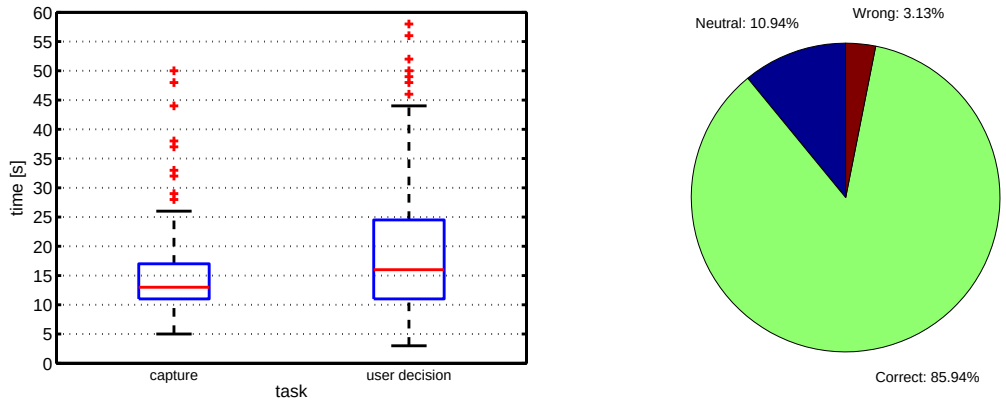

Figure 6.20: Temporal effort (left) and accuracy of user decisions (right) with the updated prototype. Holograms can be assessed by the system in approx. 15 s with all decisions being correct. Users additionally need approx. 20 s for assessment, being correct in 85.94% of all cases.

A subsequent decision by the user takes another 20 s ($M = 20.07, SD = 15.17$, see Figure 6.20). One-way within subjects ANOVAs revealed no significant effect of hologram on capture time, but on decision time ($F(7, 184) = 2.46$, $p = 0.0196$). Multiple pairwise post-hoc comparisons with Bonferroni correction for hologram revealed that the decision time for hologram 1-o ($M = 27, SD = 19.99$) was significantly different to those of hologram 1-f ($M = 12.83, SD = 14.07$). The system was able to assess the hologram correctly in all cases. Users were unsure about the validity of the hologram in 10.94% of all cases and succeeded to give a correct decision in 85.94% of all cases (see Table 6.2 for details on individual holograms).

Four users pointed out that they enjoyed using the application ('steep learning curve', 'liked it'). Half of the users mentioned that it was not completely intuitive to use the application (e.g., 'complicated', 'needs practice'). One user suggested to use textual hints or a virtual example. Another user suggested to use the wire-frame of a 3D object for alignment or to augment a half-dome on top of the element. Two users mentioned issues with deciding on the validity of a hologram (e.g., 'not clear, when patches are different', 'different colors are irritating').

Discussion: The modified selection of reference views leads to a reasonable checking time of 15 s when using the system. Due to the fact that three reference views were used,

Hologram	**1-o**	**1-f**	**2-o**	**2-f**	**3-o**	**3-f**	**4-o**	**4-f**
Correct [%]	75.0	100.0	83.3	50.0	91.7	95.8	91.7	100.0
Neutral [%]	25.0	0.0	8.3	37.5	8.3	0.0	8.3	0.0
Wrong [%]	0.0	0.0	8.4	12.5	0.0	4.2	0.0	0.0

Table 6.2: User decisions for hologram inspection using originals and substitutes.

instead of two, this is an encouraging result, which confirms that the actual selection is critical to the efficiency of the process.

While it is interesting to note that the actual decision time (20 s) takes longer than image capture, this is not relevant for the evaluated holograms, since the system was able to give a perfect decision in all cases. The significant difference in decision time between Hologram 1-o and hologram 1-f (substitute) is very likely due to a larger visual difference for this pair regarding the original and the substitute. The lower accuracy achieved by the users (85.94%) gives room for speculation that they cannot intuitively assess the evaluated holograms by themselves. Users, in particular, had issues assessing hologram 2-f correctly, which is a rotated version of the original element. They were also rather unsure about the validity of Hologram 1-o, where the patterns are subject to a larger amount of color noise.

Many participants pointed out that the interface was complicated to use. This is due to the complexity of the task, which requires simultaneous monitoring of several parameters and rather fine-grained navigation. This could be improved by using textual instructions or animation throughout the process. Using a wireframe-based alignment approach does not seem reasonable in the light of prior results (see Section 6.3.4.6).

It must be noted that the aforementioned selection of reference views, although natural for the user and beneficial regarding efficiency, may not be possible for an arbitrary security element. The reason is that the complete set of reference patterns does not necessarily become visible when recording with a flash-enabled mobile device and following the suggested path for orientation change (i.e., tilting downwards). Consequently, there is a need for specially designed security elements which allow the aforementioned selection of viewing directions. This can be considered a realistic demand, since there are already elements on the market which approximately have this property.

6.5 Conclusion and Future Work

Accounting for the importance of user guidance in mobile settings, and, in particular, for hologram verification, we conducted a series of experiments involving different user interfaces supporting manual and automatic comparison of patches.

View Alignment We first proposed a novel user guidance approach suitable for view alignment based on iron sights and the virtual horizon. This allows the user to efficiently match the pose of the phone with pre-recorded views of the hologram. We implemented this approach within a mobile AR framework for document inspection and conducted a user study comparing a digital manual with the AR system. The obtained results prove that it is feasible to capture and verify holograms in a manual setting on off-the-shelf mobile phones. However, it is necessary to decrease physical and cognitive strain for the user.

Efficient User Interfaces With the goal to improve the efficiency of the overall approach, we implemented and evaluated several different AR user interfaces for checking holograms. Alignment, constrained navigation and hybrid approaches using automatic matching were compared in a user study. Although the hybrid interface had the fastest completion time, users preferred the constrained navigation interface over the other two according to the comments received. Although this led to a considerable reduction in effort for the user, the system was not able to match the performance of users inspecting the captured images manually.

User-Friendly Parametrization As the required temporal effort as well as the accuracy called for further improvement, we modified the spatial distribution of reference views in order to mimic the typical behavior of users observed during document verification and also changed the similarity measure for patch-matching. An evaluation within a user study turned out, that hologram capture can be done in approximately 15 s. An automatic decision by the system follows immediately. Consequently, the selection of reference views is critical for the efficiency of the process. Contrary to decisions on validity made by the users, the system proved to be correct in all cases.

Future Work In order to allow the use of the application without a training phase (e.g., download from an app-store), it seems reasonable to provide an in-app tutorial through visual and textual hints for each step of the process.

It must be noted that the initial focus operation required for adjusting the camera lens to the small operating distance for capturing hologram patches my require several seconds with current off-the-shelf devices. With better control over the camera, a suitable distance could be set automatically due to the available document and pose information. This would further reducing the required temporal effort and improve the overall usability.

From the results obtained in our studies, it is evident that security elements should be designed with mobile verification by human operators in mind. Besides placing relevant appearances along main viewing directions, it would also be reasonable to support a continuous assessment of capture conditions during the process (i.e., flashlight dominance). This could be realized by using a hologram made of several parts. However, it seems beneficial to switch to a different sampling pattern in this case (i.e., orthogonal movements). Such a pattern has been commercially adopted for product protection involving a 2D barcode equipped with a unique hologram[1]. However, this solution does not support a thorough monitoring of capture conditions and uses a non-interactive setup with discrete sampling positions.

[1] `http://www.authenticvision.com`

7 Conclusion

Contents

7.1 Summary of Results

In this work, we considered the use of off-the-shelf mobile devices for document inspection in the context of AR. Initially, basic building blocks for the creation of a mobile information system for assessing documents, including those containing personal data, were established. We proposed a sequential approach, consisting of document detection and subsequent classification of the document from a rectified input image. In order to allow application of off-the-shelf mobile devices, efficient algorithms for the detection of perspectively distorted rectangular regions and purely client-side mobile visual search were proposed, implemented and confirmed through extensive evaluation to be suitable for use on ordinary devices.

Having an advanced mobile image acquisition device at hand, we went on to investigate how it could be used to support the actual process of checking security documents. To this end, we designed an efficient approach for detection and recognition of machine-readable zones, without requiring accurate alignment of the imaging device and the document. This solution, while giving more freedom to the user and providing an instant feedback channel, was shown to offer reasonable performance and accuracy, despite having to deal with the challenges of mobile setups. We also contributed a synthetic database, which can be used for further research in the field.

With the Mobile AR setup established in the initial steps, we first investigated the feasibility of mobile hologram detection and verification. For repeatable image capture of hologram patterns, a dominant light-source is required (e.g., built-in flashlight), which

limits the application mostly to indoor scenarios. While we confirmed the feasibility of those steps in an extensive evaluation, the requirement of aligning the image device with the document several times in order to get suitable data for verification, lead to impractical temporal effort and load for the user. Automatic image capture and matching along with several different user interfaces were introduced in order to increase efficiency. The outcome was a semi-automatic system for mobile hologram verification. A final evaluation taking into account typical user behavior and involving substitute holograms, revealed that it is possible to capture relevant information in approximately 15 s. Automatic decisions made by the system were able to surpass the judgment given by the actual human operator.

7.2 Lessons Learned

During the work described in this thesis, additional insights were gained that could be useful when considering further research in the current or a related field. In the following, these will be briefly discussed.

Computer Vision on Mobile Devices: With the task of processing identity documents, it became evident that state-of-the art solutions for detection and tracking are too restrictive for our goals. Although it is understandable that companies want to protect their achievements, we can speculate that slightly more flexible toolkits for Mobile AR would be beneficial for research.

Considering the issue of document classification, our experience shows there is still potential to improve upon the accuracy or latency of commercial solutions, while using only the computing power available in a handheld device. However, during various phases of this work, we were plagued with throttling of the mobile device. Consequently, monitoring the efficiency of the application is still mandatory, because peak performance is only available for a relatively short amount of time, due to thermal or energy issues.

We experienced an omnipresent lack of representative data for training and testing of algorithms involving documents. Although it is absolutely necessary to protect privacy, it seems also reasonable to demand public authorities to help in establishing anonymous databases, which can be used for the evaluation of algorithms for document verification.

Interaction and User Experience: The complex task of mobile hologram verification shows the importance of user interface design. Comparing the final efficiency achieved with results along the path, it seems very reasonable to observe the natural behavior of users when designing approaches for user guidance and interaction. If possible by any means, the intuitive behavior of users shall be determined and exploited for the achievement of the actual task. The verbal preference we observed for an actually slower interface suggest that users do not like to be patronized. This observation is increasingly important in the context of app-stores, where software is not expected to arrive with lengthy explanations

on usage. However, there is also a solid basis on computer vision required, which can cope with the desired freedom of the user.

7.3 Outlook

Especially for mobile document inspection, improving the performance of visual tracking is desirable. While this could be treated by extending the scope of features used for tracking (e.g., text), typical deformations of documents should also be handled, which, given the limited processing power available, is a challenge on its own. For mobile hologram verification, processing curved documents could mean that the element is no longer planar. This calls for additional research about the effect of these deformations on the appearance of the element. When also the front camera of a mobile device is available, gaze tracking could become a source of information to trigger the display of augmented information on the document without moving the device or the document.

As pointed out initially, the focus of fraud is shifting away from the document to the actual people handing over the document. In order to catch imposters, mobile verification should be extended to support personnel with the verification of the identity of a person in question. This could be done by automatic facial recognition, possibly also using biometric information stored in a microchip of the document (ePassport). Again, this functionality must be realized under computational constraints, despite strict requirements regarding robustness and the protection of sensitive data.

A

List of Acronyms

ANN...**A**rtificial **N**eural **N**etwork

ANOVA...**AN**alysis **O**f **VA**riance

AR...**A**ugmented **R**eality

BOW...**B**ag **O**f **W**ords

BRIEF...**B**inary **R**obust **I**ndependent **E**lementary **F**eatures

BRISK...**B**inary **R**obust **I**nvariant **S**calable **Ke**ypoints

CCD...**C**harge-**C**oupled **D**evice

CV...**C**omputer **V**ision

GPS...**G**lobal **P**ositioning **S**ystem

HD...**H**igh **D**efinition

HOG...**H**istogram **O**f **G**radients

HMD...**H**ead **M**ounted **D**isplay

LED...**L**ight-**E**mitting **D**iode

MRZ...**M**achine **R**eadable **Z**one

MSER...**M**aximally **S**table **E**xtremal **R**egions

NCC...**N**ormalized **C**ross **C**orrelation

NFC...**N**ear **F**ield **C**ommunication

NFT...**N**atural **F**eature **T**racking

OCR...**O**ptical **C**haracter **R**ecognition

ORB...**O**riented FAST and **R**otated **B**RIEF

OVD...**O**ptically **V**ariable **D**evice

PCA...**P**rincipal **C**omponent **A**nalysis

RANSAC...**R**andom **S**ample **C**onsensus

ROI...**R**egion **O**f **I**nterest

SAD...**S**um of **A**bsolute **D**ifferences

SIFT...**S**cale-**I**nvariant **F**eature **T**ransform

SSIM...**S**tructural **S**imilarity **I**ndex

SURF...**S**peeded **U**p **R**obust **F**eatures

SVM...**S**upport **V**ector **M**achine

TF-IDF...**T**erm **F**requency, **I**nverse **D**ocument **F**requency

VIZ...**V**isual **I**nspection **Z**one

SLAM...**S**imultaneous **L**ocalization and **M**apping

SVBRDF...**S**patially **V**arying **B**idirectional **R**eflectance **D**istribution **F**unction

SWT...**S**troke **W**idth **T**ransform

UX...**U**ser **E**xperience

VGA...**V**ideo **G**raphics **A**rray

B

Study on Document Capture

We conducted an experiment in order to gain more insights about the typical behavior of laypersons when recording documents with mobile devices for the purpose of inspection. In the first part of the experiment, participants were asked to record a self-made sample of an ID-document using the Samsung Galaxy S5 smartphone for as long as they deemed appropriate. During recording, the document was tracked and the pose and video information was logged onto the device including the corresponding timestamps. Afterward, participants were asked about what they had been looking for on the document. In the second part, they were asked to look specifically at the hologram through the mobile device within two trials. In the first case, users were asked to record the hologram with the document in hand, while in the second case, the document was placed on a table. In order to avoid learning effects, we balanced the order of trials among the participants.

In total, 20 participants (three female) took part in the study, which on average lasted for about ten minutes. We analyzed the obtained data by producing a map of the *unprojection* of the camera center onto the template. However, no distinct hot-spots besides the hologram became visible. After the analysis of the comments obtained by the users, it turned out, that people looked at the hologram, face, text and other specialized elements, but did not necessarily keep the element visible in the middle of the screen. However, the hologram and the face image were usually examined before all the other elements (see Figure B.1). Over all participants, the document was sampled on average for around 37 s ($M = 37.15, SD = 16.21$).

During both trials of hologram inspection, users looked at the hologram and started to tilt the document or the device. On average, users were sampling the hologram in these scenarios for around 33 s ($M = 33.35, SD = 13.59$). Changes in orientation in general took place roughly along the vertical and horizontal axes. However, there is a notable difference in behavior, depending on whether the document is in hand or on the table. While in the first case, mainly vertical movements are made into both directions, in the second case orientation changes take place in the lower direction and to the side (less distinct). The latter seems reasonable, since otherwise the user would move the screen

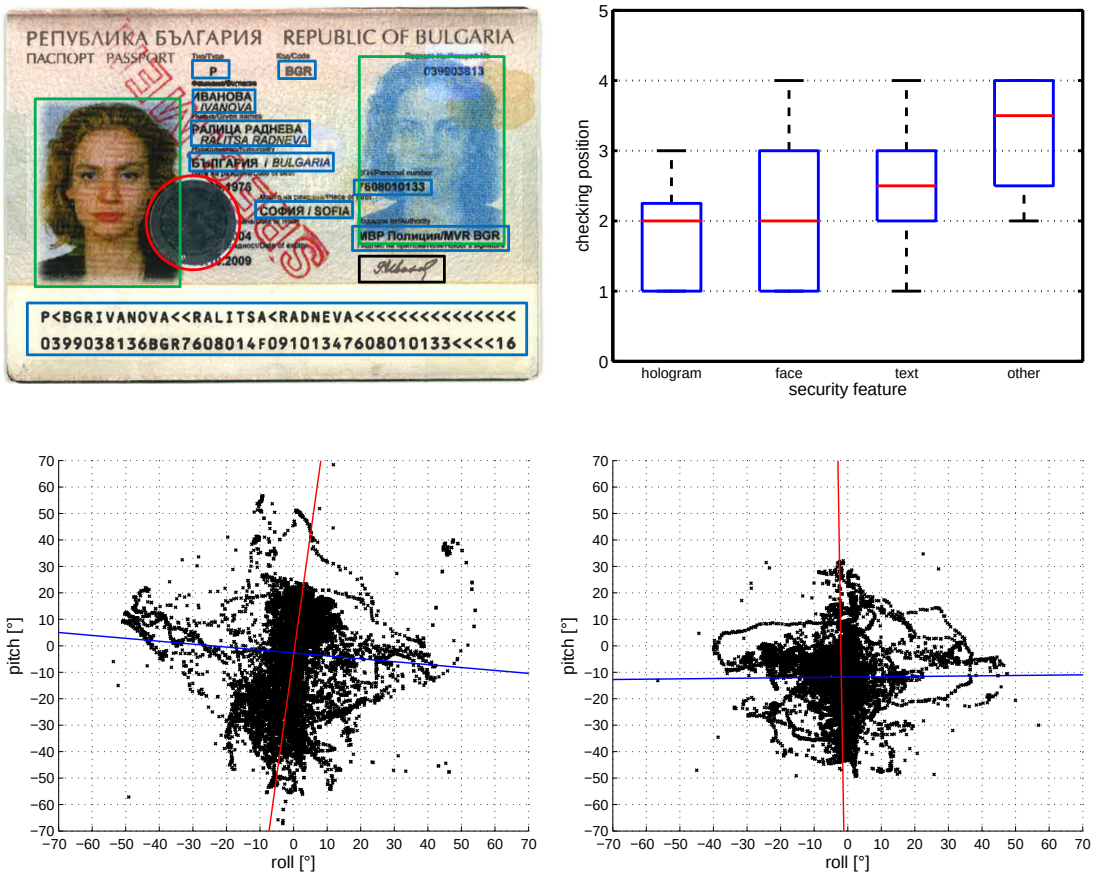

Figure B.1: Checking order of various security features for an exemplary ID-document (top row). The hologram and the face image are the most interesting elements, followed by textual information including the MRZ. Orientation changes during hologram inspection were considered with the document held in hand (bottom-left) and with the document kept on a table (bottom-right). This corresponds to tilting the document roughly in the vertical and also in the horizontal direction. In the first case, users did not move the document exactly in the vertical direction. This can also be seen by the visualized Eigenvectors (red and blue lines).

of the device away from the field of vision. It must be noted that in the first case, the majority of users tried to fix the device in one hand and tilted the document only. From the visualization of the corresponding Eigenvectors it is evident, that users did not move the document exactly in the vertical direction when holding it in hand, but also rotated it slightly.

Bibliography

[1] Ahmed, Z., Yasmin, S., Nahidul Islam, M., and Ahmed, R. U. (2014). Image processing based feature extraction of bangladeshi banknotes. In *International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, pages 1–8. (page 17)

[2] Alahi, A., Ortiz, R., and Vandergheynst, P. (2012). Freak: Fast retina keypoint. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 510–517. (page 23)

[3] Álvaro Gonzalez, Bergasa, L. M., Torres, J. J. Y., and Bronte, S. (2012). Text location in complex images. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, pages 617–620. (page 25)

[4] Arth, C., Reitmayr, G., and Schmalstieg, D. (2013). Full 6dof pose estimation from geo-located images. In *Proceedings of Asian Conference on Computer Vision (ACCV)*, ACCV'12, pages 705–717, Berlin, Heidelberg. Springer-Verlag. (page 16)

[5] Arth, C., Wagner, D., Klopschitz, M., Irschara, A., and Schmalstieg, D. (2009). Wide area localization on mobile phones. In *Proceedings of IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 73–82. (page 16)

[6] Augereau, O., Journet, N., Vialard, A., and Domenger, J.-P. (2014). Improving classification of an industrial document image database by combining visual and textual features. In *IAPR International Workshop on Document Analysis Systems (DAS)*, pages 314–318. (page 23)

[7] Azuma, R. T. (1997). A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385. (page 1, 13, 14)

[8] b. Tao, W., w. Tian, J., and Jian, L. (2002). A new approach to extract rectangular building from aerial urban images. In *International Conference on Signal Processing*, volume 1, pages 143–146. (page 22, 31)

[9] Bae, S., Agarwala, A., and Durand, F. (2010). Computational rephotography. *ACM Trans. Graph.*, 29(3):24:1–24:15. (page 27)

[10] Bataineh, B., Abdullah, S. N. H. S., and Omar, K. (2011). An adaptive local binarization method for document images based on a novel thresholding method and dynamic windows. *Pattern Recognition Letters (PRL)*, 32(14):1805–1813. (page 72)

[11] Bay, H., Ess, A., Tuytelaars, T., and Gool, L. V. (2008). Speeded-up robust features (surf). *Computer Vision and Image Understanding (CVIU)*, 110(3):346–359. (page 18, 22)

[12] Bessmeltsev, V., Bulushev, E., and Goloshevsky, N. (2011). High-speed ocr algorithm for portable passport readers. In *International Conference on Computer Graphics and Vision (GraphiCon)*. (page 19)

[13] Bhaskar, H., Werghi, N., and Al Mansoori, S. (2010). Combined spatial and transform domain analysis for rectangle detection. In *Conference on Information Fusion (FUSION)*, pages 1–7. (page 21)

[14] Bimber, O. (2004). Combining optical holograms with interactive computer graphics. *IEEE Computer*, 37(1):85–91. (page 20)

[15] Bimber, O., Zeidler, T., Grundhoefer, A., Wetzstein, G., Moehring, M., Knoedel, S., and Hahne, U. (2005). Interacting with augmented holograms. In *SPIE Practical Holography XIX: Materials and Applications*, volume 5742, pages 41–54. (page 20)

[16] Biocca, F., Tang, A., Owen, C., and Xiao, F. (2006). Attention funnel: Omnidirectional 3d cursor for mobile augmented reality platforms. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1115–1122. (page 27)

[17] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer. (page 18, 26)

[18] Bruna, A., Farinella, G. M., Guarnera, G. C., and Battiato, S. (2013). Forgery detection and value identification of euro banknotes. *Sensors*, 13(2):2515–2529. (page 18)

[19] Buraga-Lefebvre, C., Coëtmellec, S., Lebrun, D., and Özkul, C. (2000). Application of wavelet transform to hologram analysis: three-dimensional location of particles. *Optics and Lasers in Engineering*, 33(6):409–421. (page 19)

[20] Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). Brief: binary robust independent elementary features. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 778–792. (page 22)

[21] Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 8(6):679–698. (page 32)

[22] Caudell, T. and Mizell, D. (1992). Augmented reality: an application of heads-up display technology to manual manufacturing processes. In *Proceedings of the Hawaii International Conference on System Sciences*, volume ii, pages 659–669. (page 1)

[23] Cesarini, F., Lastri, M., Marinai, S., and Soda, G. (2001). Encoding of modified x-y trees for document classification. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1131–1136. (page 24)

[24] Chandrasekhar, V.and Takacs, G., Chen, D. M., Tsai, S., Reznik, Y.and Grzeszczuk, R., and Girod, B. (2012). Compressed histogram of gradients: A low-bitrate descriptor. *International Journal of Computer Vision (IJCV)*, 96(3):384–399. (page 36)

[25] Chandrasekhar, V. R., Chen, D. M., Tsai, S. S., Cheung, N.-M., Chen, H., Takacs, G., Reznik, Y., Vedantham, R., Grzeszczuk, R., Bach, J., and Girod, B. (2011). The stanford mobile visual search data set. In *MMSys*, pages 117–122. (page 39)

[26] Chang, F., Chen, C.-J., and Lu, C.-J. (2004). A linear-time component-labeling algorithm using contour tracing technique. *Computer Vision and Image Understanding (CVIU)*, 93(2):206–220. (page 32, 56)

[27] Chen, D. M., Tsai, S. S., Chandrasekhar, V., Takacs, G., Vedantham, R., Grzeszczuk, R., and Girod, B. (2010). Inverted index compression for scalable image matching. In *IEEE DCC*, page 525. (page 23)

[28] Chen, G.-H., Yang, C.-L., Po, L.-M., and Xie, S.-L. (2006). Edge-based structural similarity for image quality assessment. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 2. (page 84)

[29] Chen, S., He, Y., Sun, J., and Naoi, S. (2012). Structured document classification by matching local salient features. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, pages 653–656. (page 23)

[30] Chintamani, K., Cao, A., Ellis, R., and Pandya, A. (2010). Improved telemanipulator navigation during display-control misalignments using augmented reality cues. *Systems, Man and Cybernetics, Part A: Systems and Humans*, 40(1):29–39. (page 27)

[31] Choi, E., Lee, J., and Yoon, J. (2006). Feature extraction for bank note classification using wavelet transform. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, pages 934–937, Washington, DC, USA. IEEE Computer Society. (page 18)

[32] Comaniciu, D. and Meer, P. (2002). Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(5):603–619. (page 76)

[33] Cuthbertson, M. (2010). The changing global dynamic of travel document fraud. In *Symposium and Exhibition on ICAO MRTDs, Biometrics and Security Standards.* (page 5)

[34] Dong, Y., Wang, J., Tong, X., Snyder, J., Lan, Y., Ben-Ezra, M., and Guo, B. (2010). Manifold bootstrapping for svbrdf capture. In *ACM SIGGRAPH 2010 papers*, pages 98:1–98:10. (page 19)

[35] Donoser, M., Arth, C., and Bischof, H. (2007). Detecting, tracking and recognizing license plates. In *Proceedings of Asian Conference on Computer Vision (ACCV)*, pages 447–456, Berlin, Heidelberg. Springer-Verlag. (page 25)

[36] Dubuisson, M.-P. and Jain, A. (1994). A modified hausdorff distance for object matching. In *Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision amp; Image Processing., Proceedings of the 12th IAPR International Conference on*, volume 1, pages 566–568. (page 86)

[37] Duda, R. and Hart, E. (1972). Use of the hough transformation to detect lines and curves in pictures. *Comm. ACM*, 15(1):11–15. (page 32)

[38] E.C.B. (2013). Biannual inform. on euro banknote counterfeiting. (page 95)

[39] Epshtein, B., Ofek, E., and Wexler, Y. (2010). Detecting text in natural scenes with stroke width transform. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2963–2970. (page 25)

[40] Erol, B., Antúnez, E. R., and Hull, J. J. (2008). Hotpaper: multimedia interaction with paper using mobile phones. In El-Saddik, A., Vuong, S., Griwodz, C., Bimbo, A. D., Candan, K. S., and Jaimes, A., editors, *Proceedings of the ACM International Conference on Multimedia*, pages 399–408. ACM. (page 24)

[41] European Central Bank (ECB) (2015). Biannual information on euro banknote counterfeiting. https://www.ecb.europa.eu/press/pr/date/2015/html/pr150123.en.html . (page 3)

[42] Evans, C. (2009). Notes on the opensurf library. Technical Report CSTR-09-001, University of Bristol. (page 38)

[43] Fabrizio, J., Cord, M., and Marcotegui, B. (2009a). Text extraction from street level images. In *CMRT*, pages 199–204. (page 25)

[44] Fabrizio, J., Marcotegui, B., and Cord, M. (2009b). Text segmentation in natural scenes using toggle-mapping. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pages 2349–2352. (page 56)

[45] Feiner, S., MacIntyre, B., Hollerer, T., and Webster, A. (1997). A touring machine: prototyping 3d mobile augmented reality systems for exploring the urban environment. In *International Symposium on Wearable Computers (ISWC)*, pages 74–81. (page 14)

[46] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM*, 24(6):381–395. (page 23)

[47] Fitzmaurice, G. W. (1993). Situated information spaces and spatially aware palmtop computers. *Commun. ACM*, 36(7):39–49. (page 14)

[48] Fragoso, V., Gauglitz, S., Zamora, S., Kleban, J., and Turk, M. (2011). Translatar: A mobile augmented reality translator. In *IEEE Workshop on Applications of Computer Vision (WACV)*, pages 497–502. (page 26)

[49] Gao, H., Rusinol, M., Karatzas, D., and Llados, J. (2014). Embedding document structure to bag-of-words through pair-wise stable key-regions. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, pages 2903–2908. (page 23)

[50] Gariup, M. (2015). How to detect document and identity fraud? Seminar on ID Theft. (page 8)

[51] Gariup, M. and Soederlind, G. (2013). Document fraud detection at the border: Preliminary observations on human and machine performance. In *European Intelligence and Security Informatics Conference (EISIC)*, pages 231–238. (page 7, 28)

[52] Gasparini, S. and Bertolino, P. (2013). Stereo camera tracking for mobile devices. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 14–19. (page 16)

[53] Girod, B., Chandrasekhar, V., Chen, D. M., Cheung, N.-M., Grzeszczuk, R., Reznik, Y. A., Takacs, G., Tsai, S. S., and Vedantham, R. (2011). Mobile visual search. *IEEE Signal Processing Magazine*, 28(4):61–76. (page 36)

[54] Gomez, L. and Karatzas, D. (2014). Mser-based real-time text detection and tracking. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, pages 3110–3115. (page 26)

[55] Gruber, L., Hartl, A., Arth, C., Hauswiesner, S., and Schmalstieg, D. (2011). Rapid reconstruction of small objects on mobile phones. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 20–27. (page 17)

[56] Gschwandtner, M., Stolc, S., and Daubner, F. (2014). Optical security document simulator for black-box testing of (abc) systems. In *IEEE Joint Intelligence and Security Informatics Conference, (JISIC)*, pages 300–303. (page 7)

[57] Haindl, M. and Filip, J. (2013). *Visual Texture*. Advances in Computer Vision and Pattern Recognition. Springer Verlag. (page 77)

[58] Hart, S. G. and Staveland, L. E. (1988). *Human Mental Workload*, chapter Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. North Holland Press, Amsterdam. (page 96, 111)

[59] Hartl, A., Arth, C., and Schmalstieg, D. (2014a). Ar-based hologram detection on security documents using a mobile phone. In Bebis, G., Boyle, R., Parvin, B., Koracin, D., McMahan, R., Jerald, J., Zhang, H., Drucker, S., Kambhamettu, C., El Choubassi, M., Deng, Z., and Carlson, M., editors, *Advances in Visual Computing*, volume 8888 of *Lecture Notes in Computer Science (LNCS)*, pages 335–346. Springer International Publishing. (page 68)

[60] Hartl, A., Arth, C., and Schmalstieg, D. (2015a). Real-time detection and recognition of machine-readable zones with mobile devices. In *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)*, pages 79–87. (page 55, 59, 60)

[61] Hartl, A., Grubert, J., Reinbacher, C., Arth, C., and Schmalstieg, D. (2015b). Mobile user interfaces for efficient verification of holograms. In *Proceedings of the IEEE Virtual Reality Annual International Symposium (VR)*. (page 89)

[62] Hartl, A., Grubert, J., Schmalstieg, D., and Reitmayr, G. (2013). Mobile interactive hologram verification. In *Proceedings of IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 75–82. (page 68, 89, 102, 114)

[63] Hartl, A. and Reitmayr, G. (2012). Rectangular target extraction for mobile augmented reality applications. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, pages 81–84. (page 30)

[64] Hartl, A., Schmalstieg, D., and Reitmayr, G. (2014b). Client-side mobile visual search. In *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)*, pages 125–132. (page 30)

[65] Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition. (page 14, 21, 38)

[66] Hasanuzzaman, F., Yang, X., and Tian, Y. (2011). Robust and effective component-based banknote recognition by surf features. In *Annual Wireless and Optical Communications Conference (WOCC)*, pages 1–6. (page 18)

[67] Hassenzahl, M., Burmester, M., and Koller, F. (2003). AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität. In *Mensch & Computer 2003: Interaktion in Bewegung*, pages 187–196, Stuttgart, Germany. B. G. Teubner. (page 96, 111)

[68] He, J., Feng, J., Liu, X., Cheng, T., Lin, T.-H., Chung, H., and Chang, S.-F. (2012). Mobile product search with bag of hash bits and boundary reranking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3005–3012. (page 36)

[69] Heger, S., Portheine, F., Ohnsorge, J. A. K., Schkommodau, E., and Radermacher, K. (2005). User-interactive registration of bone with a-mode ultrasound. *Engineering in Medicine and Biology Magazine, IEEE*, 24(2):85–95. (page 27, 105)

[70] Henze, N., Schinke, T., and Boll, S. (2009). What is that? object recognition from natural features on a mobile phone. In *Workshop on Mobile Interaction with The Real World*. (page 37)

[71] Hirschmuller, H. and Scharstein, D. (2009). Evaluation of stereo matching costs on images with radiometric differences. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31(9):1582–1599. (page 84)

[72] Hollerer, T., Feiner, S., and Pavlik, J. (1999). Situated documentaries: embedding multimedia presentations in the real world. In *International Symposium on Wearable Computers (ISWC)*, pages 79–86. (page 15)

[73] Hu, J., Kashi, R., and Wilfong, G. (1999). Document classification using layout analysis. In *1999. Proceedings of the International Workshop on Database and Expert Systems Applications*, pages 556–560. (page 58)

[74] ICAO (2008). Machine readable travel documents. (page 5, 53, 59, 65)

[75] Iwamura, M., Kobayashi, T., and Kise, K. (2011). Recognition of multiple characters in a scene image using arrangement of local features. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1409–1413. (page 26)

[76] Iwamura, M., Kobayashi, T., Matsuda, T., and Kise, K. (2013). Recognition of layout-free characters on complex background. In *IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 740–741. (page 26)

[77] Jachnik, J., Newcombe, R. A., and Davison, A. J. (2012). Real-time surface light-field capture for augmentation of planar specular surfaces. In *Proceedings of IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 91–97. (page 19, 77)

[78] Janucki, J. and Owsik, J. (2003). A wiener filter based correlation method intended to evaluate effectiveness of holographic security devices. *Optics Communications*, 218(4-6):221–228. (page 20)

[79] Ji, R., Duan, L.-Y., Chen, J., Yao, H., Rui, Y., Chang, S.-F., and Gao, W. (2011). Towards low bit rate mobile visual search with multiple-channel coding. In *ACM MM*, pages 573–582. (page 36)

[80] Jung, C. and Schramm, R. (2004). Rectangle detection based on a windowed hough transform. In *Symposium on Computer Graphics and Image Processing*, pages 113–120. (page 21)

[81] Karthikeyan, S., Jagadeesh, V., and Manjunath, B. S. (2013). Learning bottom-up text attention maps for text detection using stroke width transform. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*. (page 26)

[82] Kasar, T. and Ramakrishnan, A. G. (2012). Multi-script and multi-oriented text localization from scene images. In *Proceedings of the International Conference on Camera-Based Document Analysis and Recognition (CBDAR)*, pages 1–14, Berlin, Heidelberg. Springer-Verlag. (page 25)

[83] Kato, H. and Billinghurst, M. (1999). Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proceedings of the IEEE and ACM International Workshop on Augmented Reality*, pages 85–94. (page 15)

[84] Khashman, A., Sekeroglu, B., and Dimililer, K. (2005). Deformed banknote identification using pattern averaging and neural networks. In *Proceedings of the WSEAS International Conference on Computational Intelligence, Man-machine Systems and Cybernetics*, CIMMACS'05, pages 233–237, Stevens Point, Wisconsin, USA. World Scientific and Engineering Academy and Society (WSEAS). (page 18)

[85] Klein, G. and Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. In *Proceedings of IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Nara, Japan. (page 16)

[86] Klein, G. and Murray, D. (2009). Parallel tracking and mapping on a camera phone. In *Proceedings of IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Orlando. (page 16)

[87] Kobayashi, T., Iwamura, M., Matsuda, T., and Kise, K. (2013). An anytime algorithm for camera-based character recognition. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1140–1144. (page 26)

[88] Kolev, K., Tanskanen, P., Speciale, P., and Pollefeys, M. (2014). Turning mobile phones into 3d scanners. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3946–3953. (page 17)

[89] Kosecka, J. and Zhang, W. (2002). Efficient computation of vanishing points. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 223–228. (page 21)

[90] Kunze, K., Utsumi, Y., Shiga, Y., Kise, K., and Bulling, A. (2013). I know what you are reading: Recognition of document types using mobile eye tracking. In *International Symposium on Wearable Computers (ISWC)*, ISWC '13, pages 113–116, New York, NY, USA. ACM. (page 24)

[91] Kurz, D. and Benhimane, S. (2011). Gravity-aware handheld augmented reality. In *Proceedings of IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, ISMAR '11, pages 111–120, Washington, DC, USA. IEEE Computer Society. (page 16)

[92] Kwon, H.-J. and Park, T.-H. (2007). An automatic inspection system for hologram with multiple patterns. In *SICE*, pages 2663–2666. (page 20)

[93] Lagunovsky, D. and Ablameyko, S. (1999). Straight-line-based primitive extraction in grey-scale object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 20(10):1005–1014. (page 22)

[94] Leutenegger, S., Chli, M., and R., S. (2011). Brisk: Binary robust invariant scalable keypoints. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 2548–2555. (page 23, 34)

[95] Lewis, J. R. (1991). Psychometric evaluation of an after-scenario questionnaire for computer usability studies: The asq. *SIGCHI Bull.*, 23(1):78–81. (page 111)

[96] Li, Q. (2014). A geometric framework for rectangular shape detection. *IEEE Transactions on Image Processing (TIP)*, 23(9):4139–4149. (page 22)

[97] Lin, C., Huertas, A., and Nevatia, R. (1994). Detection of buildings using perceptual grouping and shadows. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 62–69. (page 22, 31)

[98] Liu, X. (2008). A camera phone based currency reader for the visually impaired. In *Proceedings of the International ACM SIGACCESS Conference on Computers and Accessibility*, Assets '08, pages 305–306, New York, NY, USA. ACM. (page 18)

[99] Liu, X. and Doermann, D. (2008). Mobile retriever: Access to digital documents from their physical source. *International Journal on Document Analysis and Recognition (IJDAR)*, 11(1):19–27. (page 24)

[100] Liu, X., Lu, K., , and Wang, W. (2012). Effectively localize text in natural scene images. In *Proceedings of International Conference on Pattern Recognition (ICPR)*. (page 25)

[101] Liu, Y., Ikenaga, T., and Goto, S. (2007). An mrf model-based approach to the detection of rectangular shape objects in color images. *Signal Processing*, 87(11):2649–2658. (page 22)

[102] Liu, Z. and Sarkar, S. (2008). Robust outdoor text detection using text intensity and shape features. In *Proceedings of International Conference on Pattern Recognition (ICPR)*. (page 25)

[103] Lohweg, V., Hoffmann, J., Dörksen, H., Hildebrand, R., Gillich, E., Hofmann, J., and Schaede, J. (2014). Authentication of security documents and mobile device to carry out the authentication. WO Patent App. PCT/IB2014/058,776. (page 18)

[104] Loomis, J. M., Golledge, R. G., Klatzky, R. L., Speigle, J. M., and Tietz, J. (1994). Personal guidance system for the visually impaired. In *Proceedings of the Annual ACM Conference on Assistive Technologies*, Assets '94, pages 85–91, New York, NY, USA. ACM. (page 14)

[105] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110. (page 22)

[106] Mandridake, C., Ouddan, A., Hoarau, M., and Win-Lime, K. (2014). Towards fully automatic id document frauds detection. In *Workshop Interdisciplinaire sur la Sécurité Globale (WISG)*. (page 19)

[107] Matas, J., Chum, O., Urban, M., and Pajdla, T. (2002). Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of British Machine Vision Conference (BMVC)*. (page 76)

[108] Matsuda, T., Iwamura, M., and Kise, K. (2014). Performance improvement in local feature based camera-captured character recognition. In *IAPR International Workshop on Document Analysis Systems (DAS)*, pages 196–201. (page 26)

[109] McAuley, E., Duncan, T., and Tammen, V. V. (1989). Psychometric properties of the intrinsic motivation inventory in a competitive sport setting: A confirmatory factor analysis. *Research quarterly for exercise and sport*, 60(1):48–58. (page 96, 99, 111)

[110] Merino-Gracia, C., Lenc, K., and Mirmehdi, M. (2012). A head-mounted device for recognizing text in natural scenes. In *Proceedings of the International Conference on Camera-Based Document Analysis and Recognition (CBDAR)*, pages 29–41, Berlin, Heidelberg. Springer-Verlag. (page 25)

[111] Miao, L. and Peng, S. (2006). Perspective rectification of document images based on morphology. In *International Conference on Computational Intelligence and Security*, volume 2, pages 1805–1808. (page 21)

[112] Micusik, B., Wildenauer, H., and Kosecka, J. (2008). Detection and matching of rectilinear structures. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (page 22)

[113] Milgram, P. and Kishino, F. (1994). A taxonomy of mixed reality visual displays. *IEICE Transactions on Information and Systems*, pages 1321–1329. (page 1, 13)

[114] Milyaev, S., Barinova, O., Novikova, T., Kohli, P., and Lempitsky, V. S. (2013). Image binarization for end-to-end text understanding in natural images. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 128–132. IEEE Computer Society. (page 25)

[115] Minetto, R., Thome, N., Cord, M., Fabrizio, J., and Marcotegui, B. (2010). Snoopertext: A multiresolution system for text detection in complex visual scenes. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pages 3861–3864. (page 25)

[116] Minetto, R., Thome, N., Cord, M., Stolfi, J., Precioso, F., Guyomard, J., and Leite, N. J. (2011). Text detection and recognition in urban scenes. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 227–234. (page 25)

[117] Mishra, A., Alahari, K., and Jawahar, C. V. (2012). Top-down and bottom-up cues for scene text recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (page 25)

[118] Moffat, A. and Anh, V. N. (2005). Binary codes for non-uniform sources. In *IEEE DCC*, pages 133–142. (page 39)

[119] Möhring, M., Lessig, C., and Bimber, O. (2004). Video see-through ar on consumer cell-phones. In *Proceedings of IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 252–253. (page 16)

[120] Nakai, T., Kise, K., and Iwamura, M. (2005). Hashing with local combinations of feature points and its application to camera-based document image retrieval. In *Proceedings of the International Conference on Camera-Based Document Analysis and Recognition (CBDAR)*, pages 87–94. (page 24)

[121] Neumann, L. and Matas, J. (2011). Text localization in real-world images using efficiently pruned exhaustive search. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 687–691. IEEE. (page 25)

[122] Neumann, L. and Matas, J. (2012). Real-time scene text localization and recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3538–3545. (page 25)

[123] Nister, D. and Stewenius, H. (2006). Scalable recognition with a vocabulary tree. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2161–2168. (page 23)

[124] Ojala, T., Pietikainen, M., and Harwood, D. (1994). Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, volume 1, pages 582–585. (page 26)

[125] Opitz, M., Diem, M., Fiel, S., Kleber, F., and Sablatnig, R. (2014). End-to-end text recognition using local ternary patterns, mser and deep convolutional nets. In *IAPR International Workshop on Document Analysis Systems (DAS)*, pages 186–190. (page 26)

[126] Pal, S., Blumenstein, M., and Pal, U. (2011). Automatic off-line signature verification systems: A review. *IJCA Proceedings on International Conference and workshop on Emerging Trends in Technology (ICWET)*, pages 20–27. (page 19)

[127] Pan, Q., Arth, C., Reitmayr, G., Rosten, E., and Drummond, T. (2011a). Rapid scene reconstruction on mobile phones from panoramic images. In *Proceedings of IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 55–64. (page 17)

[128] Pan, Y.-F., Hou, X., and Liu, C.-L. (2011b). A hybrid approach to detect and localize texts in natural scene images. *IEEE Transactions on Image Processing (TIP)*, 20(3):800–813. (page 26)

[129] Park, T.-H. and Kwon, H.-J. (2010). Vision inspection system for holograms with mixed patterns. In *IEEE Conference on Automation Science and Engineering (CASE)*, pages 563–567. (page 20)

[130] Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572. (page 39)

[131] Pilu, M. (2001). Extraction of illusory linear clues in perspectively skewed documents. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 363–368. (page 21)

[132] Pirchheim, C. and Reitmayr, G. (2011). Homography-based planar mapping and tracking for mobile phones. In *Proceedings of IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 27–36. (page 16, 21)

[133] Pitkäaho, T. and Naughton, T. J. (2009). Stereo vision based approach for extracting features from digital holograms. In Osten, W. and Kujawinska, M., editors, *Fringe 2009*, pages 1–6. Springer Berlin Heidelberg. (page 19)

[134] Pock, T., Urschler, M., Zach, C., Beichel, R., and Bischof, H. (2007). A duality based algorithm for tv-l1-optical-flow image registration. In *Proceedings of the Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 511–518. (page 97)

[135] Pramila, A., Keskinarkaus, A., Rahtu, E., and Seppänen, T. (2011). Watermark recovery from a dual layer hologram with a digital camera. In Heyden, A. and Kahl, F., editors, *Proceedings of Scandinavian Conference on Image Analysis (SCIA)*, pages 146–155. Springer. (page 19)

[136] Prisacariu, V., Kahler, O., Murray, D., and Reid, I. (2013). Simultaneous 3d tracking and reconstruction on a mobile phone. In *Proceedings of IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 89–98. (page 17)

[137] Radványi, M., Solymár, Z., Stubendek, A., and Karacs, K. (2011). Mobile banknote recognition: Topological models in scene understanding. In *Proceedings of the International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL)*, pages 185:1–185:5, New York, NY, USA. ACM. (page 18)

[138] Reitmayr, G. and Drummond, T. (2006). Going out: Robust model-based tracking for outdoor augmented reality. In *Proceedings of IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 109–118, Washington, DC, USA. IEEE Computer Society. (page 16)

[139] Reitmayr, G. and Schmalstieg, D. (2001). Mobile collaborative augmented reality. In *International Symposium on Augmented Reality (ISAR)*, pages 114–123. (page 15)

[140] Rekimoto, J. and Nagao, K. (1995). The world through the computer: Computer augmented interaction with real world environments. In *Proceedings of the Annual ACM Symposium on User Interface and Software Technology*, UIST '95, pages 29–36, New York, NY, USA. ACM. (page 15)

[141] Ren, P., Wang, J., Snyder, J., Tong, X., and Guo, B. (2011). Pocket reflectometry. In *Proceedings of the International Conference on Computer graphics and interactive techniques (SIGGRAPH)*, SIGGRAPH '11, pages 45:1–45:10, New York, NY, USA. ACM. (page 19)

[142] Roy, A., Halder, B., Garain, U., and Doermann, D. (2015). Machine-assisted authentication of paper currency: an experiment on indian banknotes. *International Journal on Document Analysis and Recognition (IJDAR)*, pages 1–15. (page 18)

[143] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). Orb: an efficient alternative to sift or surf. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 2564–2571. (page 23)

[144] Saoi, T., Goto, H., and Kobayashi, H. (2005). Text detection in color scene images based on unsupervised clustering of multi-channel wavelet features. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 690–694. IEEE Computer Society. (page 25)

[145] Schöps, T., Engel, J., and Cremers, D. (2014). Semi-dense visual odometry for AR on a smartphone. In *Proceedings of IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. (page 16)

[146] Shafait, F., Keysers, D., and Breuel, T. (2008). Efficient implementation of local adaptive thresholding techniques using integral images. In *SPIE Conference on Document Recognition and Retrieval (DRR)*. SPIE. (page 32, 70)

[147] Shaw, D. and Barnes, N. (2006). Perspective rectangle detection. In *Workshop on Applications of CV at ECCV*. (page 21)

[148] Shingu, J., Rieffel, E., Kimber, D., Vaughan, J., Qvarfordt, P., and Tuite, K. (2010). Camera pose navigation using augmented reality. In *Proceedings of IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 271–272. (page 28)

[149] Sivic, J. and Zisserman, A. (2003). Video google: a text retrieval approach to object matching in videos. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 1470–1477. (page 23)

[150] Snavely, N., Seitz, S. M., and Szeliski, R. (2006). Photo tourism: Exploring photo collections in 3d. *ACM Trans. Graph.*, 25(3):835–846. (page 27)

[151] Soukup, D., Štolc, S., and Huber-Mörk, R. (2015). Analysis of optically variable devices using a photometric light-field approach. In *Proc. SPIE 9409, Media Watermarking, Security, and Forensics.* (page 20)

[152] Sukan, M., Elvezio, C., Oda, O., Feiner, S., and Tversky, B. (2014). Parafrustum: Visualization techniques for guiding a user to a constrained set of viewing positions and orientations. In *Proceedings of the Annual ACM Symposium on User Interface Software and Technology (UIST)*, pages 331–340, New York, NY, USA. ACM. (page 28)

[153] Sun, Q., Lu, Y., and Sun, S. (2010). A visual attention based approach to text extraction. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, pages 3991–3995. (page 26)

[154] Sun, S.-Y., Gilbertson, M. W., and Anthony, B. W. (2013). Computer-guided ultrasound probe realignment by optical tracking. In *IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 21–24. (page 27)

[155] Sutherland, I. E. (1968). A head-mounted three-dimensional display. In *Proceedings of AFIPS*, pages 757–764. (page 1, 13)

[156] Takeda, K., Kise, K., and Iwamura, M. (2012). Real-time document image retrieval on a smartphone. In *IAPR International Workshop on Document Analysis Systems (DAS)*, pages 225–229. (page 24)

[157] Tanskanen, P., Kolev, K., Meier, L., Camposeco, F., Saurer, O., and Pollefeys, M. (2013). Live metric 3d reconstruction on mobile phones. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 65–72. (page 17)

[158] Tarjan, R. E. (1972). Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160. (page 57)

[159] Thomas, B., Close, B., Donoghue, J., Squires, J., De Bondi, P., Morris, M., and Piekarski, W. (2000). Arquake: an outdoor/indoor augmented reality first person application. In *International Symposium on Wearable Computers (ISWC)*, pages 139–146. (page 15)

[160] Thomas, B. H., Demczuk, V., Piekarski, W., Hepworth, D., and Gunther, B. (1998). A wearable computer system with augmented reality to support terrestrial navigation. In *International Symposium on Wearable Computers (ISWC)*, pages 168–171, Pittsburgh, USA. IEEE Computer Society. (page 15)

[161] Toyama, T., Suzuki, W., Dengel, A., and Kise, K. (2013). User attention oriented augmented reality on documents with document dependent dynamic overlay. In *Proceedings of IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 299–300. (page 24)

[162] Trzcinski, T., Christoudias, M., Fua, P., and Lepetit, V. (2013). Boosting binary keypoint descriptors. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2874–2881. (page 23)

[163] Tsai, S., Chen, D. M., Takacs, G., Chandrasekhar, V., Vedantham, R., Grzeszczuk, R., and Girod, B. (2010). Fast geometric re-ranking for image-based retrieval. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pages 1029–1032. (page 23)

[164] Tsai, S. S., Chen, D., Takacs, G., Chandrasekhar, V., Singh, J. P., and Girod, B. (2009). Location coding for mobile image retrieval. In *MMCC*, pages 8:1–8:7. (page 36)

[165] Tsai, S. S., Chen, H., Chen, D. M., and Girod, B. (2014). Word-hogs: Word histogram of oriented gradients for mobile visual search. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pages 3968–3972. (page 23)

[166] Tsai, S. S., Chen, H., Chen, D. M., Vedantham, R., Grzeszczuk, R., and Girod, B. (2011). Mobile visual search using image and text features. In *Asilomar Conference on Signals, Systems and Computers (ACSCC)*, pages 845–849. (page 23)

[167] Usilin, S., Nikolaev, D., Postnikov, V., and Schaefer, G. (2010). Visual appearance based document image classification. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pages 2133–2136. (page 24)

[168] van Beusekom, J., Keysers, D., Shafait, F., and Breuel, T. (2006). Distance measures for layout-based document image retrieval. In *International Conference on Document Image Analysis for Libraries (DIAL)*, pages 232–242. (page 24, 72)

[169] van Renesse, R. L. (2005). *Optical Document Security*. Artech House, third edition. (page 3, 6, 7, 8)

[170] Vandenberg, S. G. and Kuse, A. R. (1978). Mental rotations, a group test of three-dimensional spatial visualization. *Perceptual and Motor Skills*, 47(2):599–604. (page 109)

[171] Ventura, J., Arth, C., Reitmayr, G., and Schmalstieg, D. (2014). Global localization from monocular slam on a mobile phone. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 20(4):531–539. (page 16)

[172] Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *International Journal of Computer Vision (IJCV)*, 57(2):137–154. (page 24)

[173] Wagner, D., Mulloni, A., Langlotz, T., and Schmalstieg, D. (2010a). Real-time panoramic mapping and tracking on mobile phones. In *Virtual Reality Conference (VR), 2010 IEEE*, pages 211–218. (page 16)

[174] Wagner, D., Reitmayr, G., Mulloni, A., Drummond, T., and Schmalstieg, D. (2010b). Real-time detection and tracking for augmented reality on mobile phones. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 16(3):355–368. (page 16, 34, 60, 69)

[175] Wagner, D. and Schmalstieg, D. (2003). First steps towards handheld augmented reality. In *International Symposium on Wearable Computers (ISWC)*, pages 127–135. IEEE Computer Society Press. (page 15)

[176] Wang, L., Katsuyama, Y., Fan, W., He, Y., Sun, J., and Hotta, Y. (2013). Text detection in natural scene images with user-intention. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pages 2256–2259. (page 26)

[177] Wang, X., Yang, M., Cour, T., Zhu, S., Yu, K., and Han, T. (2011). Contextual weighting for vocabulary tree based image retrieval. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 209–216. (page 23, 39)

[178] Wu, Y., Xue, L., Li, C., and Xiong, Z. (2007). Tdars, a fusion based ar system for machine readable travel documents. In Smith, M. and Salvendy, G., editors, *Human Interface and the Management of Information. Interacting in Information Environments*, volume 4558 of *Lecture Notes in Computer Science (LNCS)*, pages 1129–1138. Springer Berlin Heidelberg. (page 19)

[179] Yi, C. and Tian, Y. (2011). Text string detection from natural scenes by structure-based partition and grouping. *IEEE Transactions on Image Processing (TIP)*, 20(9):2594–2605. (page 26)

[180] Yin, X.-C., Hao, H.-W., Sun, J., and Naoi, S. (2011). Robust vanishing point detection for mobilecam-based documents. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 136–140. (page 21)

[181] Yonemoto, S. (2013). An interactive image rectification method using quadrangle hypothesis. In Petrosino, A., editor, *International Conference on Image Analysis and Processing (ICIAP)*, volume 8157 of *Lecture Notes in Computer Science (LNCS)*, pages 51–60. Springer. (page 21)

[182] Zhang, W. and Kosecka, J. (2003). Extraction, matching and pose recovery based on dominant rectangular structures. In *IEEE International Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis (HLK)*, pages 83–91. (page 21)

[183] Zhou, W., Lu, Y., Li, H., and Tian, Q. (2012). Scalar quantization for large scale image search. In *ACM MM*, pages 169–178. (page 36)

[184] Zhou Wang, A.C. Bovik, H. R. S. and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing (TIP)*, 13(4):600–612. (page 84)

[185] Zhu, K.-h., Qi, F.-h., Jiang, R.-j., and Xu, L. (2007). Automatic character detection and segmentation in natural scene images. *Journal of Zhejiang University SCIENCE A (JZUS)*, 8(1):63–71. (page 25)

[186] Zhu, Y., Carragher, B., Mouche, F., and Potter, C. S. (2003). Automatic particle detection through efficient hough transforms. *IEEE Trans. Med. Imaging*, 22(9):1053–1062. (page 21)

[187] Zhu, Y. and Qingzhi, Z. (2011). Rectangle detection by the chain-code tracing. In *International Conference on Electrical and Control Engineering (ICECE)*, pages 759–762. (page 22)