AR-Based Hologram Detection on Security Documents Using a Mobile Phone

Andreas Hartl, Clemens Arth, and Dieter Schmalstieg

Graz University of Technology, Austria
{hartl,arth,schmalstieg}@icg.tugraz.at

Abstract. Holograms are used frequently in creating fraud resistant security documents, such as passports, ID cards or banknotes. The key contribution of this paper is a real time method to automatically detect holograms in images acquired with a standard smartphone. Following a robust algorithm for creating a tracking target, our approach evaluates an evolving stack of observations of the document to automatically determine the location and size of holograms. We demonstrate the plausibility of our method using a variety of security documents, which are common in everyday use. Finally, we show how suitable data can be captured in a novel mobile gaming experience and draw the link between serious applications and entertainment.

1 Introduction

Holograms are a special group of view-dependent elements which exhibit large changes in appearance depending on the viewing angle and light sources in the environment. Besides water marks and specialty paper, holograms are essential features for security documents to prevent counterfeit and fraud [1]. Banknotes form a group of security documents that almost everyone has encountered. In addition, there are several other use cases such as product and brand protection, where the detection of a hologram may be sufficient to tell genuine items from fake ones.

The common approach to detect forgery - at least for banknotes - is the "Feel-Look-Tilt" test, which has to be done manually to verify the authenticity of a document and requires reference information (e.g., a printed manual). Naturally this is not reasonable in situations pressed for time, like at a cashier desk for example. However, holograms are also used for passports and ID cards. For those documents, personnel employed at airports or public facilities have to be trained for detecting counterfeit and fake documents. This causes significant cost and still cannot prevent errors by the human operator due to fatigue. Holograms are considered highly fraud-resistant; therefore, the development of automatic tools and algorithms for detection and verification deserves interest in research.

Most related work in the context of holograms is focused on automatic validation and reconstruction using camera images and Computer Vision (CV) algorithms. In [2] the wavelet transform is used to compute a digital reconstruction of particle holograms . A Wiener filter is used in [3] to quantitatively assess

G. Bebis et al. (Eds.): ISVC 2014, Part II, LNCS 8888, pp. 335–346, 2014.

hologram quality. Automatic inspection systems for holograms using sets of patterns illuminated with multiple LEDs on a hemisphere are proposed in [4, 5]. In [6] the watermark of a dual-layer hologram placed on uniform background is detected and read using a stationary, well-aligned image acquisition setup with controlled illumination. Different algorithms for capturing holograms related to the spatially varying bidirectional reflectance distribution function (SVBRDF) were proposed in [7–9]. Related work for sample acquisition and assisted verification with the help of Augmented Reality (AR) can be found in [10].

Common to all relevant approaches is that the hologram location is given as prior knowledge. The main contribution of this work 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. To the best of our knowledge, we are the first to investigate this task in more detail. The presented algorithm runs in real time on standard smartphone hardware.

The necessity of sampling the appearance of holograms from multiple view points for detection and verification 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.

In Section 2 we describe the approach used to create and to track the document and to perform frame selection and registering the stack of observations. In Section 3 we describe the algorithm for extracting the hologram area, and present experimental results in Section 4, alongside with a mobile prototype application. The proposed approach and the obtained results are discussed in Section 5, and concluding remarks are given in Section 6.

2 Document Detection, Tracking and Registration

Security documents are usually made of paper or paperboard and are generally of rectangular shape. For reasons of robustness and efficiency, we limit ourselves to roughly planar regions. Recording such documents with mobile devices is a difficult task due to varying personal data on the document, viewpoint changes, illumination, unexpected user behavior and limitations of mobile camera hardware. Consequently, several frames should be evaluated for increased robustness, which can be accomplished by using a mobile AR setup.

2.1 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 detection of perspectively distorted rectangles, running in real time on smartphones, which serves as a basic building block. While the basics are described in

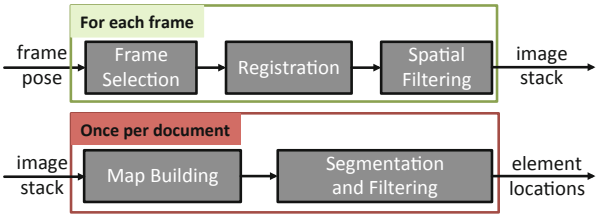


Fig. 1. Illustration of the required steps for hologram detection per frame and per document/evaluation of the image stack

brief here, the reader is referred to [11] for a detailed description and evaluation of the approach.

The user is asked to place the camera in front of a document and to trigger the detection. Given a predefined region of interest (ROI) in an input image, an edge map is computed using the Canny edge detector with automatic threshold selection. Regions containing text-like structures are filtered to remove a large amount of noise, followed by the detection of lines using the Hough transform. The lines detected are grouped according to their rough direction. The initial hypotheses for a rectangular region are formed by considering pairs of line bundles, containing four lines in total. Through intersection of lines and assuring that the intersection points fall into the ROI, the number of hypotheses can be significantly reduced. A final ranked list of rectangle hypotheses is generated by calculating a support function on a dilated edge image. The top candidate of the list is chosen and a homography is calculated to create a rectified representation. The dimensions of the rectified image are determined by averaging the pixel width and height of the chosen hypothesis.

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 [12]. Harris corners and normalized cross correlation (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 and therefore to save 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.

2.2 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, generally more viewpoints should be recorded. Based on the results of the target tracking module, our algorithm

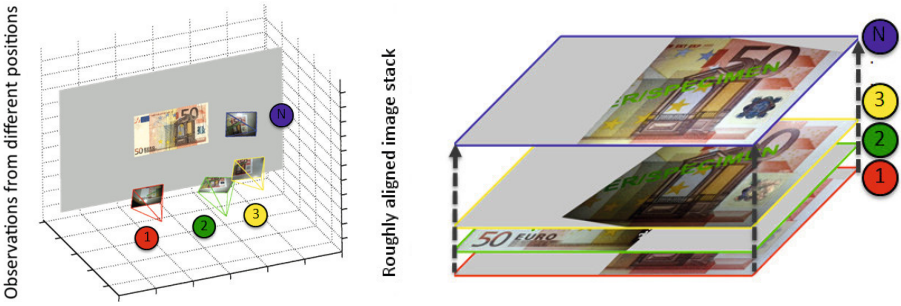


Fig. 2. The target is tracked and observations from different positions are recorded. Through the estimated homography, each image is rectified and pushed onto the stack.

consists of three main parts to create an image stack: (i) frame selection, (ii) warping/registration and (iii) spatial filtering (see Figure 1).

2.3 Frame Selection and Registration

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.

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 therefore generates a stack of equally sized pictures (see Figure 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 actual camera image and the observations may undergo significant perspective distortion. The rectified images are therefore 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.

2.4 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

similar to the approach proposed in [13]. We account for incomplete image information (black areas in warping) by recording valid areas used in filtering in a second mask.

3 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 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.

3.1 Map Building

We treat the stack of measurements for each position (x, y) as a temporal series. 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 3(a)). We tested using the mean m_0 or the median m_1 along with the average quadratic error in image space

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

or the average absolute error

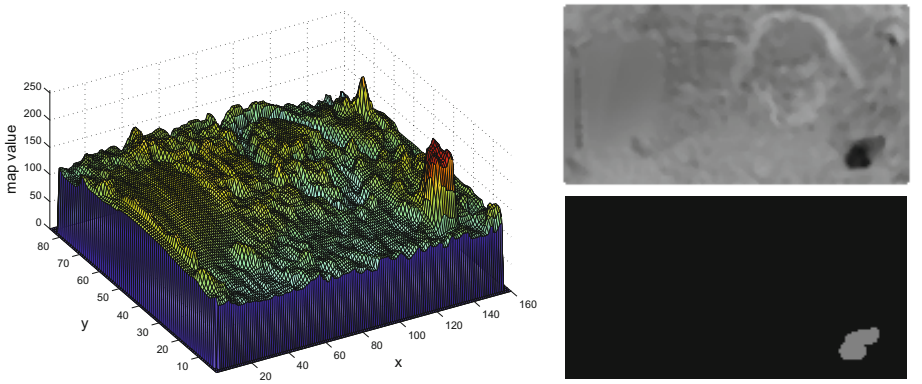
$$e_1(x,y) = \frac{1}{L(x,y)} \sum_{l=1}^{L(x,y)} |v_l(x,y) - m|, \tag{2}$$

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.

3.2 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



(a) (b)

Fig. 3. (a) surface plot of a hologram map obtained from a sample document. (b) corresponding scaled intensity image, and selection result using adaptive thresholding.

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 [14] (see Figure 3(b) for an exemplary result). In order to save runtime, integral images are used for filtering.

The computed regions are then 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.

4 Experiments

We recorded several documents with holograms right on the 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 (see Figure 4).

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

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 a relatively high number of frames (90) in order to allow a more detailed evaluation of the algorithm. However, this is not a problem since the document can be tracked with more than 30 FPS on the recording device. Document regions are warped to have a

Fig. 4. Left: Using Google Glass to perform hologram detection on a foreign passport. Right: Augmented detection results of exemplary samples using in our evaluation.

maximum extension of 160 pixels and spatial filtering is carried out on a 3x3 window.

4.1 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 [15]. Similar to layout distance metrics for documents, our metric has to account for missing or superfluous elements, but we only consider overlapping regions, relating a region present in the ground-truth mask R_{gt} to the regions R_j obtained by our detection approach, where Ov denotes the number of overlapping pixels and s_{ov} the obtained score,

$$s_{ov}(R_{gt}, R_j) = \frac{2Ov(R_{gt}, R_j)}{area(R_{gt}) + area(R_j)} \quad . \tag{3}$$

We heuristically determined $s_{ov} \geq 0.4$ as a good threshold to treat a detection as true positive. 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.

We then ran the proposed hologram detection algorithms using different algorithmic combinations for analyzing the image stack (see Figure 5). 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 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. It must also be noted that hologram detection across the entire document

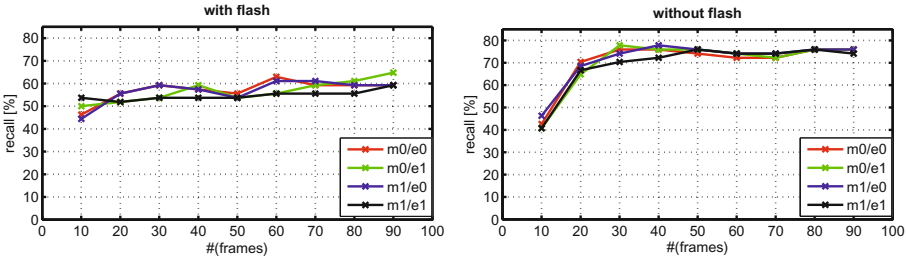


Fig. 5. Results of hologram detection using 14 different documents when binarizing the hologram map with flash enabled (left) and flash disabled (right). Contrary to hologram verification, not using the flashlight gives better results. There are only small changes when more than 30 frames are used.

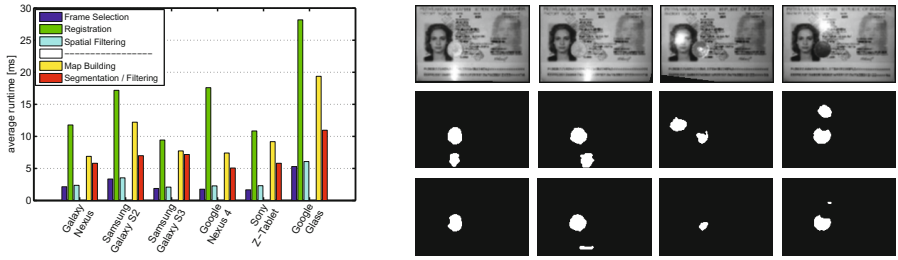


Fig. 6. Left: Runtime for the individual parts of our approach on several different devices. Note that the first group of tasks is done once per frame, while the second group of algorithms needs only be done once per session. Right: Segmentation of single stack layers. Input image (top row), MSER regions (middle row), MSER regions from modified input image (highlight detection, inpainting).

is a very different task compared to the verification of a single hologram, where the flashlight is beneficial for capturing all significant views [10].

4.2 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 6, the individual algorithmic parts take between 13-25 ms per-frame on most devices. Warping using the available pose information requires most of the time. Final evaluation of the hologram map including segmentation

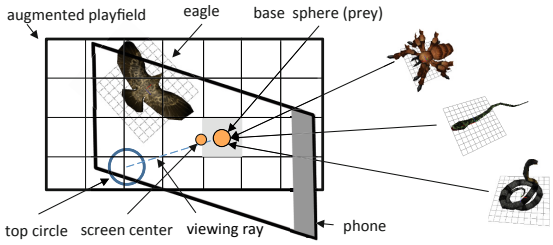


Fig. 7. Mobile AR game concept. Cells are augmented on the document for prey placement. The sighting device consists of a base sphere, top circle and a focus point on the screen. For alignment (triggering the eagle to dive for prey), all elements must roughly coincide.

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 4).

4.3 Mobile AR Game

From our experience, the need to sample documents from multiple viewpoints is not apparent to laymen. However, we observed that in some mobile AR games, players already follow a similar motion pattern. Consequently, we consider an application of the proposed image acquisition approach as background task in a game context. This could increase document security as a byproduct of playing mobile games, when the overall goal is not the verification of a document, but personal entertainment.

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 similar to [10]. However, instead of requiring detailed alignment, we use loose matching of the center and viewing direction and do not consider the in-plane rotation of the viewing ray. Cells (base point), polar angle and azimuthal angle of the ray are randomly selected for placing prey (see Figure 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 8).

We evaluated the game running the proposed hologram detector in the background within a user study taking place in several offices around our department. 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

Fig. 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.

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 (image capture for detection finished) or by the game (maximum number of rounds (32)).

The data captured by 7 of 10 users during gameplay was suitable for hologram detection. Additionally, another suitable hologram map was recorded, which did not make it through final verification (mean M=0.75, stddev SD=0.40). Participants played for around two minutes (M=98.9 s, SD=46.6 s). 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-point 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) with 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.

5 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 [16] or Mean-Shift [17]. 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 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.

6 Conclusion

In this paper, 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. The experimental results presented proof of the plausibility of the algorithms proposed. An evaluation within a mobile game gave encouraging results towards improving document security while playing.[1]

In future work we will carry out a more elaborate stack analysis to better detect holograms with a small number of views, in particular on documents with shiny surfaces. The detection of holograms is only the first step towards automatic testing and verification of holograms as embedded security features. As a next step we want to investigate approaches to perform automatic hologram

[1] A submission video can be found at `http://youtu.be/XzCkbc1GNME`

verification along with methods to guide the user in the process of efficient data acquisition and counterfeit detection.

References

1. Van Renesse, R.: Optical document security. Optoelectronics Library. Artech House (2005)
2. Buraga-Lefebvre, C., Coëtmellec, S., Lebrun, D., Özkul, C.: Application of wavelet transform to hologram analysis: three-dimensional location of particles. Optics and Lasers in Engineering 33, 409–421 (2000)
3. Janucki, J., Owsik, J.: A Wiener filter based correlation method intended to evaluate effectiveness of holographic security devices. Optics Communications 218, 221–228 (2003)
4. Kwon, H.J., Park, T.H.: An automatic inspection system for hologram with multiple patterns. In: SICE, Annual Conference, pp. 2663–2666 (2007)
5. Park, T.H., Kwon, H.-J.: Vision inspection system for holograms with mixed patterns. In: IEEE Conference on Automation Science and Engineering (CASE), pp. 563–567 (2010)
6. Pramila, A., Keskinarkaus, A., Rahtu, E., Seppänen, T.: Watermark recovery from a dual layer hologram with a digital camera. In: Heyden, A., Kahl, F. (eds.) SCIA 2011. LNCS, vol. 6688, pp. 146–155. Springer, Heidelberg (2011)
7. Ren, P., Wang, J., Snyder, J., Tong, X., Guo, B.: Pocket reflectometry. ACM Trans. Graph. 30, 45:1–45:10 (2011)
8. Dong, Y., Wang, J., Tong, X., Snyder, J., Lan, Y., Ben-Ezra, M., Guo, B.: Manifold bootstrapping for svbrdf capture. In: ACM SIGGRAPH, 98:1–98:10 (2010)
9. Jachnik, J., Newcombe, R.A., Davison, A.J.: Real-time surface light-field capture for augmentation of planar specular surfaces. In: ISMAR, pp. 91–97 (2012)
10. Hartl, A., Grubert, J., Schmalstieg, D., Reitmayr, G.: Mobile interactive hologram verification. In: ISMAR, pp. 75–82 (2013)
11. Hartl, A., Reitmayr, G.: Rectangular target extraction for mobile augmented reality applications. In: Proceedings of the International Conference on Pattern Recognition, pp. 81–84 (2012)
12. Wagner, D., Reitmayr, G., Mulloni, A., Drummond, T., Schmalstieg, D.: Real-time detection and tracking for augmented reality on mobile phones. TVCG 16, 355–368 (2010)
13. Shafait, F., Keysers, D., Breuel, T.: Efficient implementation of local adaptive thresholding techniques using integral images. In: DRR, SPIE (2008)
14. Bataineh, B., Abdullah, S.N.H.S., Omar, K.: An adaptive local binarization method for document images based on a novel thresholding method and dynamic windows. Pattern Recogn. Lett. 32, 1805–1813 (2011)
15. van Beusekom, J., Keysers, D., Shafait, F., Breuel, T.: Distance measures for layout-based document image retrieval. In: Int. Conference on Document Image Analysis for Libraries (DIAL), vol. 11, p. 242 (2006)
16. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: BMVC, 36.1–36.10 (2002)
17. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. PAMI 24, 603–619 (2002)