Developmental Analysis of a Markerless Hybrid Tracking Technique for Mobile Augmented Reality Systems

Waqas Khalid Obeidy(✉), Haslina Arshad(✉), Siok Yee Tan, and Hameedur Rahman

Center for Artificial Intelligence Technology, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia
waquaskhalid@yahoo.com, haslinarshad@ukm.edu.my, esthertan90@hotmail.com, rhameedur@gmail.com

Abstract. Continuous tracking in Augmented Reality (AR) applications is essential for registering and augmenting the digital content on top of the real world. However, tracking on handheld devices such as PDAs or mobile phones enforces many restrictions and challenges in the form of efficiency and robustness which are the standard performance measures of tracking. This work focuses on the pre-analysis required for the development of an Accelero-Visual Markerless Hybrid Tracking Technique. The technique combines visual feature based tracking with the accelerometer sensor of the smartphones to make the process of tracking more efficient and robust. Pre-Analysis is performed for the visual and sensor based tracking approaches required to design the hybrid tracking technique. For visual tracking, the best keypoint detector and descriptors are analyzed. Careful selection of these visual tracking elements during the analysis stage helps in achieving much efficient and robust markerless augmented reality tracking results on a modern day smartphone.

Keywords: Markerless tracking · Mobile augmented reality · Keypoint detection · Computer vision

1 Introduction

Early Augmented Reality (AR) systems were based on desktop computer and custom input output devices and backpacks with Head Mounted Displays (HMD). With the passage of time, the trend of display in AR has switched from the backpack with HMD to low cost small handheld device such as monitors, PDA's, Smartphone. Augmented Reality (AR) requires real-time tracking to track the users or device position in order to register it in respect to the real world. Augmented Reality (AR) and Virtual Reality (VR) require real-time and accurate 6DOF pose tracking of devices such as head-mounted displays and tangible interface objects. Tracking on handheld devices such as PDAs or mobile phones enforces many restrictions and challenges that are not present on stationary or PC-based setups. In marker less AR applications, the challenges that common users face are lack of memory and slow processing speed [1]. AR researchers in recent years have been working hard in order to achieve efficiency and robustness in the tracking environment of both; desktop systems and mobile systems. The process of

H. Badioze Zaman et al. (Eds.): IVIC 2015, LNCS 9429, pp. 99–110, 2015.
DOI: 10.1007/978-3-319-25939-0_9

tracking can be improved by using a markerless hybrid tracking technique. Such a system makes use of more than one kind of tracking techniques and instruments such as the sensors. Usage of inertial sensors such as accelerometer, gyroscopes and gravitational vectors can readily improve the efficiency of a feature based tracking system which uses computer vision [2]. Though there are many existing methods available for hybrid tracking in Augmented Reality, there also exist many issues on efficiency and robustness of those tracking methods especially when implemented on mobile devices. This work presents the details of the pre-analysis phase required for the development of an efficient markerless hybrid tracking technique for mobile augmented reality systems.

2 Related Works

Mobile Augmented Reality differs in many aspects from traditional mobile Augmented Reality [3]. Smartphones are inexpensive and attractive targets for the implementation of Augmented Reality but they have very limited memory and processing power as compared to PC's. Markerless tracking or tracking from natural features is a complex process and usually demands high computational power. It is therefore difficult to use robust natural feature tracking in mobile applications of Augmented Reality (AR), which runs with limited computational resources, such as on Tablet PCs [4]. Feature detection is used for different purposes and therefore performance is evaluated in terms of location, speed and accuracy. However, it is difficult to achieve the complete accuracy only through algorithms because this increases the complexity of the system and demands even higher computational power [5].

Different kinds of tracking techniques have presented by researchers can be categorized into sensor based, visual and hybrid tracking techniques. The process of tracking can be improved by using a hybrid tracking system. Such a system makes use of more than one kind of tracking techniques and instruments. The idea of combining a visual tracking system and inertial sensors is not new in augmented reality. Even before [6] came up with the idea of combining computer vision techniques with external sensors for robust and accurate orientation using a HMD device, several works have already been reported. According to [7], they successfully predicted marker position by combining fiducial markers with inertial sensors. Later in 2004 [8] showed that their work was robust enough to solve some of the matters which Azuma and Naimark were working upon. In 2010, [9] also worked on increasing the accuracy of feature detection where they found that GPS proved a good solution only in the large environments. He suggested that accuracy of the initialization process can be improved by the use of sensors.

A number of different researches were led to accomplish real-time markerless tracking on mobile phones. The most prominent of those were led by [10, 11]. The first fully self-contained markerless natural feature tracking system capable of tracking full 6 degrees of freedom (6DOF) at real-time frame rates (30 Hz) from natural features using solely the built-in camera of the phone of that time was developed [10]. He used a heavily modified version of SIFT to enable marker-based natural feature tracking on mobile phones in real-time which although allowed his system to run on real-time but with certain limitations, such as the fast motion movements and overall robustness of the system. Another good example of hybrid tracking can be found in the work of [12]. He proposed the use of a hybrid system that employs both computer vision and integrated sensors present in most new smartphones to facilitate pose estimation. Recent works of [2, 13, 14] proved that efficiency and robustness of tracking can now be improved tremendously with the help of strong inbuilt sensors available in almost every phone sold in the market today. A recent work by [15] presented a system for real world objects recognition and camera pose estimation from natural features for mobile AR. The system recognizes real world objects in real-time directly without any marker and desktop server by extracting natural features using optimized "Speed up Robust Features" SURF algorithm for mobile architecture.

Although hybrid tracking techniques are the most promising way to deal with the challenges in indoor and outdoor mobile augmented reality environments, they certainly face many challenges in terms of their applicability on today's smartphones. They [14, 16, 17] worked on integrating sensors of the smartphones along with the visual tracking proposed by [10] to develop AR interfaces and perform specific tasks of navigation. The use of sensor was specific to the kind of tasks required to be accomplished by the users and demanded intensive training. To improve the robustness of [10] work on visual tracking, authors [12] estimated three of the six degrees of freedom of pose using integrated sensors and the remaining three using feature tracking. Although he used SURF descriptor to attain real-time working system as suggested by Wagner, his system was not very robust and was susceptible to losing track of the AR environment in different tracking conditions. More importantly, the system barely met the minimum requirements of 20–30 Hz set by [10] for real-time performance. This indicates the urgency for a new markerless hybrid tracking technique for smartphones which is more efficient and robust than previous works.

3 Development Analysis

Before a visual tracking approach is carefully designed and combined with sensors, a detailed analysis of selection of suitable tracking dataset, keypoint detector, keypoint descriptors, the type of sensors and the type of Platform and Hardware used is performed. Following are the details of these analysis.

3.1 Mobile AR Tracking Dataset

For a long time, Quam's [18] Yosemite sequence used to be the reference used for evaluating optical flow algorithms. Today, the Middlebury [19] datasets are the reference for optical flow. Theoretically, these images could be used to evaluate tracking algorithms as well. However, due to the very limited number of frames/image pairs given and the completely different goal set when creating these datasets, the result from an evaluation using these datasets will be missing important factors such as e.g. motion blur and the irregular movements coming from a human camera operator. Specifically for markerless systems, some researchers have used Mikolajzyk and Zimmerman's datasets however the problem with them is that they only considers a very limited number of objects and factors influencing the tracking, e.g. the lighting conditions. Following the Zimmerman's approach, it is not possible to have reliable ground truth in the case of blurry or noisy images. It is also not possible to recover the camera position and orientation when the points used to determine the pose are not in the field of view of the camera. Consequently, the performance of the tested algorithms could not be evaluated in the presence of noise, motion blur or for some relative position between the camera and the tracked objects.

In the last few years, markerless visual tracking reached the level where a large variety of algorithms could be successfully used in a wide range of Augmented Reality applications. However markerless visual tracking lacks benchmark datasets not allowing a fair comparison between state-of-the-art computer vision algorithms. To fulfil the growing need of common objective datasets with ground truth metaio has developed a dataset that allow fast performance estimation in terms of speed and accuracy of a newly designed algorithm and its fair comparison with the existing ones [20]. Metaio identified four different types of tracking targets classified by texture richness and repeatability. Each type is represented by two targets in the dataset. Metaio also determined five standard factors that have the biggest influence on the performance of the tracking and which are related to the camera motion, the size of the tracked object in the image and the lighting conditions; one sequence per target is dedicated to each influence. Therefore, in addition to metaio, the old Mikolajczyk datasets have also been used for fair comparison with previous works during the development of our markerless hybrid tracking technique.

3.2 Sensor Tracking Analysis

Today's smartphones are incredible little machines which comes along with various sensors including the accelerometer, gyroscope, magnometer, proximity sensor, light sensor, location sensor, barometer, thermometer, pedometer, heart rate monitor finger print sensor and many more. The Android platform provides several of these sensors that help monitor the motion of a device. Two of these sensors are usually hardware-based such as the accelerometer and gyroscope, and many of the other sensors can be either hardware-based or software-based (the gravity, linear acceleration, and rotation vector sensors). The acceleration sensor measures the acceleration applied to the device, including the force of gravity. This information can help in detection of the movement of the device around its x, y and z directions and hence tell if the device is in stable condition or not. This information gathered from the accelerometer sensor can therefore control the amount of visual tracking cycles required for smooth and successful augmented reality experience saving a tremendous amount of memory on the device. It also makes the complete tracking process faster. The accelerometer provides the shake and tilting values which are usually the cause of motion, and motion blur during the process of tracking. Therefore this research relies on the usage of accelerometer to help improve and speed up the tracking process.

3.3 Platform and Hardware Analysis

For the purpose of this study, the two leading smartphone operating systems and four popular smartphones were evaluated as potential development platform. Android is a mobile operating system that is mostly open source. For Java developers, it offers a high-level application framework called Android SDK. Android apps are modular insofar as they have standard, high-level interfaces for launching each other and sharing data. Mobility, a high level of abstraction, and video processing support are features which make android devices the most suitable for mobile augmented reality applications. All the phones compared during the analysis fulfilled the basic requirements for mobile AR implementation. Although Samsung Galaxy Note's hardware specs are not the best among its competitors Iphone 5S and Galaxy Note 3 and new smartphones, but they are sufficient enough to test and run the proposed hybrid tracking technique. This especially helps in comparing the results with the previous researches which have used similar lower specs smartphone devices. Samsung Galxy Note has a Dual core, 1400 MHz, ARM Cortex-A9 processor, 1 GB RAM, a Mali-400 MP4 graphic processor. The device runs on Android OS 3.0. It has a 5.7 in. Super AMOLED capacitive touchscreen support multi touch gesture.

3.4 Visual Tracking Analysis

The most crucial step in reducing the amount of data to process and make real-time markerless tracking feasible is reliable detection and matching of features across consecutive frames. To do this, we analyzed the most important visual tracking elements that can be used. These elements include; keypoint detection, keypoint description and matching.

(a) *Preprocessing Analysis.* This section examines the time it takes to prepare the video frames that can be used for keypoint detection, description, matching and further processes involved in the development of an augmented reality system. In order to determine the average frame-rates and frame intervals for each camera mode a simple android OpenCV application was developed that measured the frames per second (fps) and the time between frames of a camera video feed in milliseconds. It is evident from the reading generated from this test summarized in Table 1, that the resolution of 640 × 480 is the most suitable resolution to work for this application since it yields the best frame rates at minimal frame intervals.

Table 1. Comparison of frame-rates and frame intervals.

Resolution	Frame rate	Frame interval
800 × 480	23 fps	43.47 ms
640 × 480	27 fps	36.8 ms
480 × 360	28 fps	35.71 ms
192 × 144	20 fps	50.8 ms

The next step is to prepare incoming video frames before passing them for different computer vision processes. One of the most important steps is converting incoming video frames to greyscale, also known as calculating image intensity. In order to determine the amount of time required to convert an incoming video frame, a simple application that measures the amount of time it takes for incoming video frames is developed. Only focused, unprepared images and video sequences are used for this step instead of prepared frames. Table 2 shows that it takes a minimal of 1.1 ms to convert a frame into greyscale at the selected 640 × 480 resolution.

Table 2. Comparison of greyscale timings.

Resolution	Greyscale time
800 × 480	1.3 ms
640 × 480	0.92 ms
480 × 360	0.95 ms
192 × 144	0.6 ms

(b) *Keypoint Detection Analysis.* Feature detection initiates the whole process of tracking by detecting the keypoints from the reference images and the scenes. These keypoints are later used by the feature descriptors to further the tracking process. The seminal work of [21] presented a comprehensive evaluation of the most competent detection methods at the time, which revealed no single all-purpose detector but rather the complementary properties of the different approaches depending on the context of the application. Many keypoint detectors include an orientation operator (SIFT and SURF are two prominent examples), but FAST does not. There are various ways to describe the orientation of a keypoint; many of these involve histograms of gradient computations, for example in [22] and the approximation by block patterns in SURF [23]. These methods are either computationally demanding, or in the case of SURF, yield poor approximations. FAST and its variants are efficient and finds reasonable corner keypoints, although it must be augmented with pyramid schemes for providing scale invariance, therefore is the best suitable option for keypoint detection for this research. FAST and its variants are the method of choice by most researchers for finding keypoints in real-time systems that match visual features [2, 10, 11]. These conclusions are further acknowledged by the practical analysis of keypoint detectors and descriptors performed on a real mobile device in the next subsection.

To configure these tests, a simple application is developed that determines the total number of keypoints found and the processing time required for the above mentioned set of keypoint detectors respectively. For most of the tests, the normal texture dataset image "ISETTA" is used because it has the best distribution of keypoints and performs best among all the other metaio dataset images. Vienna image is often used as additional dataset image for comparison with [10] work (Fig. 1). During the analysis, Pyramid

FAST and FAST finds a huge number of keypoints compared to other methods in all the four different images of the dataset. Unfortunately the keypoints found by Pyramid FAST and FAST contain a lot of noise and hence may not be suitable for further tracking. HARRIS, ORB, STAR and BRISK found respectively less but noise free keypoints. Naturally all the keypoints detectors found more number of keypoints in “WALL” and “LUCENT” images which are of High Texture and Repetitive Texture respectively.

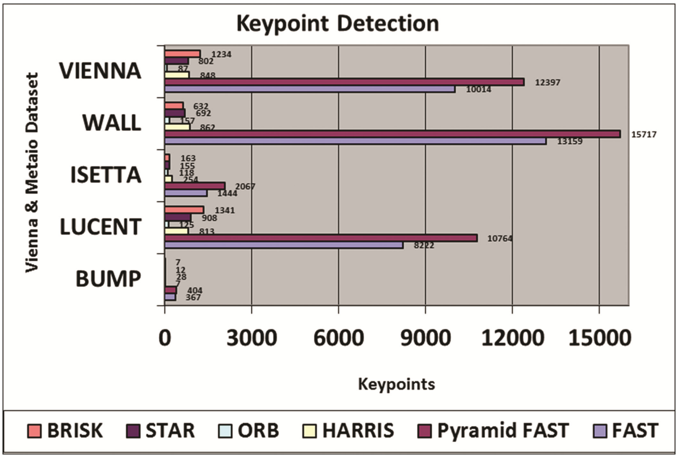

Fig. 1. Number of keypoint detected by various keypoint detectors

The speed of feature detection is tested using two criteria’s; by total amount of time spent for the detection of keypoints on the whole frame (Fig. 2). As expected, FAST detector provides best detection time per feature. When compared for the time consumed to detect the number of keypoints on the “ISETTA” image, unsurprisingly FAST and its variants such as Pyramid FAST, ORB and BRISK performed better than HARRIS and STAR detectors. As seen in the Fig. 2 HARRIS and STAR are multiple times slower than FAST and its variants. STAR took the most time to compute the keypoints and hence is not a suitable choice for the development of a real-time augmented reality application’s efficiency.

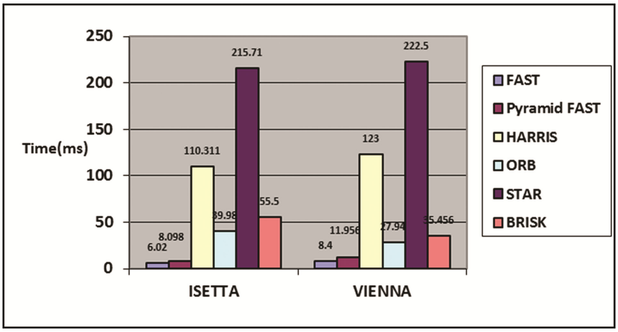

Fig. 2. Total time taken to detect all keypoints

Though FAST spent the least time to compute the keypoints, but detected hundreds of noisy scale variant keypoints which makes it unsuitable for AR. Pure Pyramid FAST does provide scale invariance by calculating FAST at different scales but it found even more keypoints than FAST and hence can make tracking unstable. However when Pyramid FAST is used in ORB, the keypoints produced are very few in number, consistent and noise free. By using Pyramid FAST and retaining only the top N matches, ORB takes slightly more time to compute the keypoints than FAST but finds more stable keypoints. BRISK is made up of AGAST detector which is another variant of FAST but consumes more than 50 ms to compute the keypoints. BRISK takes longer than other FAST variants and can hinder the tracking speed. The minimum frame rate required for the development of the proposed mobile augmented reality application is 10 fps which means 100 ms per frame [24]. Therefore the best choice among all the tested detectors PyramidFast which computes scale invariance keypoints in less than 9 ms and provides enough room for other computer vision processes such as feature description, matching and pose estimation to take place within the designated 100 ms.

(c) *Keypoint Description Analysis.* In order to identify and match keypoints across images, descriptors of the keypoints must be built. The description must be distinctive for each keypoint, but also consistent under all viewpoints. A straightforward approach is the derivation of intensity or color histogram of the local patch followed by some normalization to make it invariant to illumination changes. However, these simple descriptors are not invariant to scale, rotation and illumination. The most well-known descriptor is SIFT [25]. A 128-dimensional vector is obtained from a grid of histograms of oriented gradient. Its high descriptive power and robustness to illumination change have ranked it as the reference keypoint descriptor for the past decade. SIFT and SURF are based on histograms of gradients. These computations cost time. Even though SURF speeds up the computation using integral images, it isn't fast enough for most AR applications running on smartphones.

Binary descriptors come in handy as one can encode most of the information of a patch as a binary string using only comparison of intensity images. This can be done very fast, as only intensity comparisons need to be made. In general, binary descriptors are composed of three parts: A sampling pattern, orientation compensation and sampling pairs. Every binary descriptor has its own sampling pattern, own method of orientation calculation and its own set of sampling pairs. The authors [26] showed that it is possible to shortcut the dimensionality reduction step by directly building a short binary descriptor in which each bits are independent. BRISK and FREAK are other binary descriptors which unlike BRIEF and ORB also contain a sampling pattern. The most recent descriptor is called FREAK and it is based on a nuero-scientific research. It uses Gaussian kernels with different sizes to smooth the intensities of each sampling point.

Since many evaluation of popular descriptors such as SIFT, PCASIFT, SURF and USURF can already be found in the literature [21, 23, 27, 28], testing these descriptors would lie out of the scope for this research. Moreover as discussed in new binary descriptors such as BRIEF, ORB, BRISK and FREAK have already been proven to be much faster than SIFT and SURF and most suitable kind of descriptors for real-time image recognition applications. Hence, only binary descriptors are tested based on their description and matching time for the keypoints detected by using PyramidFAST detector. All the descriptors are matched using Brute-Force matcher.

To configure this test, a sample video at the resolution of 640 × 480 pixels is used. During the tests, only feature description and matching times are measured. Five different datasets (Isetta, Bump, Wall, Lucent, Vienna) are used to test the efficiency of binary descriptors; BRIEF, ORB, BRISK and FREAK respectively. Kruskal-Wallis test has also been performed in order to find the overall variance in terms of efficiency of the descriptors (Figs. 3 and 4).

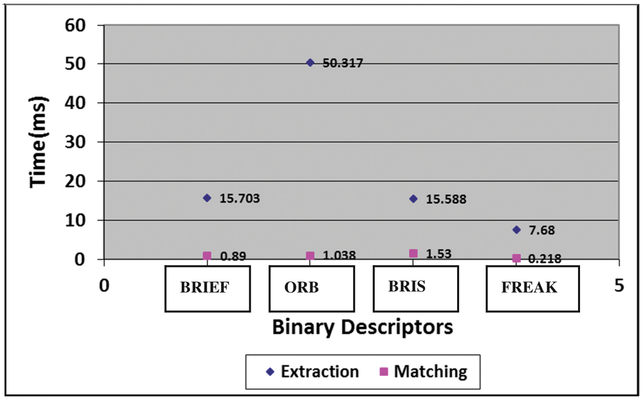

Fig. 3. Keypoint description and matching time

Kruskal-Wallis Test

Dataset/Descriptors	BRIEF	ORB	BRISK	FREAK
Isetta	15.703	50.317	15.588	7.68
Bump	11.23	42.39	11.26	6.42
Wall	18.21	58.23	17.89	7.87
Lucent	16.12	52.41	16.01	7.8
Vienna	13.10	43.12	12.98	7.21

Ranks

	B	N	Mean Rank
	BRIEF	5	10.80
	ORB	5	18.00
A	BRISK	5	10.20
	FREAK	5	3.00
	Total	20	

Test Statistics[a,b]

	A
Chi-Square	16.097
df	3
Asymp. Sig.	.001

a. Kruskal Wallis Test

b. Grouping Variable: B

```
*Nonparametric Tests: Independent Samples.
NPTESTS - CRITERIA ALPHA=0.05  CILEVEL=95.
```

Fig. 4. Readings of Kruskal-Wallis test

The graph clearly shows that descriptor matching is an instantaneous process and consumes less than 1.5 ms in most cases. The fastest matching keypoints are extracted using FREAK descriptor which unsurprisingly, also clocks the least time for keypoint description. BRIEF also performs well in the test but unfortunately is not stable and loses tracking due to its weak rotation invariance. Kruskal-Wallis test performed on different descriptors clearly shows that there is significant timing difference between them. FREAK outperforms ORB and BRISK in most of the viewpoint and photometric performance tests except of blurring. It also performs better than others during the rotation and zoom tests. FREAK outperforms all the recent state-of-the-art keypoint descriptors while remaining simple and faster with lower memory load, hence proving the most suitable choice for real-time image matching performance required for this research.

4 Results

A tracking technique would not be very effective if it detects and computes the keypoints that can be tracked only either at a fixed angle, scale or lighting. One of the most important aspects of a Visual Tracking Technique is that it must track the same points over different views of the same scene. The elements identified during the Development Analysis help us achieve both; robustness and efficiency. The technique was implemented and tested inside a mobile augmented reality application which detected markerless planar targets and rendered a cube on top of it (Fig. 5).

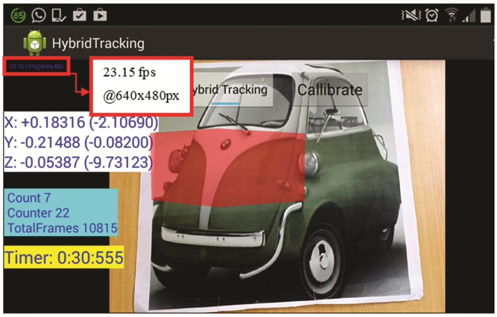

Fig. 5. Hybrid tracking technique

The Hybrid Tracking Technique allowed visual tracking to take place less frequently during a given time by allowing the sensors to take over for the remaining time. The results of the conducted efficiency and the robustness tests proved the tracking performance has been improved after the implementation of the suitable visual tracking and markerless hybrid tracking techniques. The hybrid tracking produced at least 19 Hz faster frame rates than previous researches. Moreover the robustness tests showed great improvements in all the tested sequences and overcame the limitations of rotation and scale invariance found in previous works of [10, 12, 24].

5 Conclusion

Majority of this work explores the potential of a Markerless Hybrid Tracking Technique pre-developmental analysis and tests. Achieving real-time performance, efficiency and robustness in AR are found to be the biggest challenges faced by mobile augmented reality. The technique is specially designed to cater the needs of a more efficient and robust mobile augmented reality system Therefore, the main goals and objectives of this research revolved around the study of Tracking and Mobile Augmented Reality concepts. The results at the end of the research proved that the tracking performance has been significantly improved after the implementation of the suitable visual tracking elements identified during the developmental or pre-analysis phase detailed in this work.

Acknowledgment. The authors would like to thank all those participated in this work as part of the project sponsored by research university grant Universiti Kebangsaan Malaysia (FRGS/1/2013/ICT01/UKM/02/9).

References

1. Yee, T.S., Arshad, H., Obeidy, W.K.: Car advertisement for android application in augmented reality. Int J. Inf. Syst. Eng. (online) 2(1) April 2014
2. Kurz, D., et al.: Absolute spatial context-aware visual feature descriptors for outdoor handheld camera localization. In: International Conference on Computer Vision Theory and Applications (2014)
3. Henrysson, A.: Bringing Augmented Reality to Mobile Phones. Ph.D thesis, Department of Science and Technology, The Institute of Technology, Linköping University (2007)
4. Wagner, D., Schmalstieg, D.: Making augmented reality practical on mobile phones, part 1. Comput. Graph. Appl. IEEE **29**(3), 12–15 (2009)
5. Rosenblum, L.J., et al.: The development of mobile augmented reality. In: Dill, J., Earnshaw, R., Kasik, D., Vince, J., Wong, P.C. (eds.) Expanding the Frontiers of Visual Analytics and Visualization, pp. 431–448. Springer, London (2012)
6. Azuma, R.T., et al.: Making augmented reality work outdoors requires hybrid tracking. In: Proceedings of the First International Workshop on Augmented Reality. Citeseer (1998)
7. Naimark, L., Foxlin, E.: Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. In: Proceedings of the 1st International Symposium on Mixed and Augmented Reality. IEEE Computer Society (2002)
8. Jiang, B., Neumann, U., You, S.: A robust hybrid tracking system for outdoor augmented reality. In: Proceedings of Virtual Reality. IEEE (2004)
9. Seo, B.-K., et al.: A tracking framework for augmented reality tours on cultural heritage sites. In: Proceedings of the 9th ACM SIGGRAPH Conference on Virtual-Reality Continuum and its Applications in Industry. ACM (2010)
10. Wagner, D., et al.: Real-time detection and tracking for augmented reality on mobile phones. IEEE Trans. Vis. Comput. Graph. **16**(3), 355–368 (2010)
11. Klein, G., Murray, D.: Parallel tracking and mapping on a camera phone. In: 8th IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2009. IEEE (2009)
12. Van Wyk, M.M.: The effects of Teams-Games-Tournaments on achievement, retention, and attitudes of economics education students. J. Soc. Sci. **26**(3), 183–193 (2011)
13. Siltanen, S.: Theory and applications of marker-based augmented reality, VTT Science 3 (2012). http://www.vtt.fi/inf/pdf/science/2012/S3.pdf
14. Arth, C., Mulloni, A., Schmalstieg, D.: Exploiting sensors on mobile phones to improve wide-area localization. In: 2012 21st International Conference on Pattern Recognition (ICPR). IEEE (2012)

15. Khan, U.U.H., et al.: Objects recognition and pose calculation system for mobile. Indian J. Sci. Res. Technol. **3**(1), 40–50 (2015)
16. Mulloni, A., Seichter, H., Schmalstieg, D.: Handheld augmented reality indoor navigation with activity-based instructions. In: Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services. ACM (2011)
17. Schall, C.J., et al.: Surgical instrument articulation joint cover. Google Patents (2010)
18. Heeger, D.J.: Model for the extraction of image flow. JOSA A **4**(8), 1455–1471 (1987)
19. Baker, S., et al.: A database and evaluation methodology for optical flow. Int. J. Comput. Vis. **92**(1), 1–31 (2011)
20. Lieberknecht, S., et al.: A dataset and evaluation methodology for template-based tracking algorithms. In: 8th IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2009. IEEE (2009)
21. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell. **27**(10), 1615–1630 (2005)
22. Lowe, D.G.: Object recognition from local scale-invariant features. In: The Proceedings of the Seventh IEEE International Conference on Computer Vision. IEEE (1999)
23. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
24. Wagner, D., Langlotz, T., Schmalstieg, D.: Robust and unobtrusive marker tracking on mobile phones. In: 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, ISMAR 2008. IEEE (2008)
25. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
26. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: binary robust independent elementary features. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 778–792. Springer, Heidelberg (2010)
27. Morel, J.-M., Yu, G.: ASIFT: a new framework for fully affine invariant image comparison. SIAM J. Imaging Sci. **2**(2), 438–469 (2009)
28. Miksik, O., Mikolajczyk, K.: Evaluation of local detectors and descriptors for fast feature matching. In: 2012 21st International Conference on Pattern Recognition (ICPR). IEEE (2012)