ISSN 1330-3651(Print), ISSN 1848-6339 (Online) https://doi.org/10.17559/TV-20180408120907
Original scientific paper

An Efficient Feature Extraction Scheme for Mobile Anti-Shake in Augmented Reality

Lifen ZHANG, Guangyong GAO, Caixue ZHOU, Zongmin CUI, Lihua WANG

Abstract: In recent years, augmented reality on mobile devices has become popular. Mobile shakes are the most typical type of interference in mobile augmented reality. To negate such interference, anti-shake is an urgent requirement. To enhance anti-shake efficiency, we propose an efficient feature extraction scheme for mobile anti-shake in augmented reality. The scheme directly detects corners to avoid the non-extreme constraint such that the efficiency of feature extraction is improved. Meanwhile, the scheme only updates the added corners during mobile shakes, which improves the accuracy of feature extraction. In the experiments, the memory consumption of existing methods is almost double compared to that in our scheme. Further, the runtime of our scheme is only half of the runtime of the existing methods. The experimental results demonstrate that our scheme performs better than the existing classic methods on mobile anti-shake in terms of memory consumption, efficiency, and accuracy.

Keywords: augmented reality; efficient anti-shake; feature extraction; mobile shake

1 INTRODUCTION

Augmented reality (AR) [1] is a new research field in virtual reality technology. In recent years, AR technology has been widely used in military, medical, education, culture, entertainment, and other fields [2, 3].

With the characteristics of small volume, portability, and popularity, mobiles are the most popular equipment used by people in their daily lives. Mobiles [4, 5] have become the ideal platform to achieve an AR system [6], owing to which mobile AR systems can have broad application prospects.

Mobiles are typically limited by their computational power. For a large amount of computation, feature extraction is an important factor affecting mobile systems in real time. In feature extraction, a mobile shake is the most typical event that occurs when a user operates the phone with his/her hand [7]. At this time, camera poses change slightly. Tracking the camera pose frame-by-frame decreases the AR performance in real time. Therefore, enhancing anti-shake efficiency has become a key research area for feature extraction in mobile AR [8].

To enhance mobile anti-shake efficiency, we propose an efficient feature extraction scheme for mobile anti-shake in AR. The scheme is named EAS (Efficient Anti-Shake). First, we propose a new corner judgment model to avoid the non-extreme constraint. Meanwhile, EAS uses the adjacent pixel approximation principle to reduce corner judgment complexity. The principle enhances the efficiency of feature extraction for mobile shakes. Next, we propose a mobile anti-shake algorithm that only updates new corners. On the one hand, the algorithm reduces the computational complexity for mobile shakes. On the other hand, the update is from the corner direction. Thus, the algorithm improves the accuracy of the feature extraction. The experimental results demonstrate that our scheme EAS significantly outperforms other schemes in terms of memory consumption, efficiency, and accuracy.

Our contributions are summarized as follows:

(1) We proposed a new corner judgment model to avoid the non-extreme constraint.
(2) We used the adjacent pixel approximation principle to reduce corner judgment complexity.
(3) We proposed a mobile anti-shake algorithm that only updates new corners.

The remainder of the paper is organized as follows. Section 2 describes the related works. Section 3 presents an efficient feature extraction model. Section 4 describes a mobile anti-shake algorithm. Section 5 provides the algorithm comparison. Section 6 shows the experimental results. Finally, Section 7 concludes the paper.

2 RELATED WORKS

2.1 SIFT

The scale-invariant feature transform (SIFT) algorithm [9-12] is a partial matching algorithm. It obtains key points in different scale spaces established by the Gaussian fuzzy function. The algorithm is invariant to image attributes, including rotation, translation, size, and brightness. The operator of the SIFT scheme has great judgment ability. However, the scheme requires a high computing cost, thus leading to poor real-time performance when the mobile is shaking. We propose a mobile anti-shake algorithm that only updates new corners. The algorithm reduces the computational complexity for mobile shakes. Therefore, our scheme is more efficient than SIFT.

2.2 FAST and BRIEF

The features of accelerated segment test (FAST) algorithm [13] does not produce multiscale features. In addition, FAST feature points do not contain direction information. Thus, FAST loses rotational invariance. FAST feature points are described using the binary robust independent elementary features algorithm [13]. BRIEF establishes the descriptor for these feature points. Its primary task is to describe the detected feature points. As a binary coded descriptor, it is primarily used in visual target tracking, image registration, target positioning, and image fusion. Their updating involves feature points, whereas our updating makes use of corners. Thus, our algorithm improves feature extraction accuracy. Further, our scheme is more efficient than FAST and BRIEF.

2.3 ORB

The oriented FAST and rotated BRIEF [14-17] algorithms deal with the losing rotational invariance issue

associated with FAST. Meanwhile, ORB removes the noise sensitivity problem associated with BRIEF. The feature extraction algorithm of the ORB method performs better than SIFT for mobile shakes. However, the ORB feature extraction algorithm detects the pseudo corners that require non-extreme constraints. Thus, ORB increases the computational complexity of feature extraction. We propose a new corner judgment model to avoid the non-extreme constraint. Therefore, our scheme is more efficient than ORB.

2.4 SURF

The speeded up robust features (SURF) algorithm [18] is a type of scale and rotation-invariant point feature. It extracts features from coarse granularity to fine granularity through the pyramid structure of multilayer analysis. It simplifies the scale space generation of an integral image. Therefore, it is more efficient than SIFT in feature extraction. However, it cannot fully meet the real-time requirements when it is directly applied to image tracking. We use the adjacent pixel approximation principle to reduce the complexity of corner judgment, thereby enhancing real-time performance. The principle enhances the efficiency of feature extraction for mobile shakes. Thus, our scheme is more efficient than SURF.

2.5 Other Schemes

Çalışkan et al. [19] proposed an efficient noisy pixel detection model for CT images using extreme learning machines. Several other schemes exist as well [20, 21]. However, they did not focus on mobile shake scenarios.

3 BASE FEATURE EXTRACTION

EAS includes two parts: base feature extraction and feature extraction for mobile anti-shake.

To improve the efficiency of mobile feature extraction, we propose a scale spatial corner detection algorithm that uses a segment detection method to compute the value of the anticipated-testing pixel.

Definition 1 (Corner). Given an anticipated-testing pixel p, if p synchronously satisfies the following three conditions, p is denoted as a pseudo corner, which means that p is not a corner. Otherwise, if the three conditions cannot be synchronously satisfied, p is denoted as a corner.
(1) There is a pair of pixels $\{i, i'\}$ in the circle with a radius of 3.
(2) $\{i, i'\}$ have similar pixel values with p.
(3) $\{i, i'\}$ are symmetrical about p.

To compute the direction of our corners, we first define the field moment shown in Eq. (1). In Eq. (1), (x, y) is the coordinate of a random pixel and $I(x, y)$ denotes the pixel value of (x, y). Eq. (1) defines the $(q + r)$-order moment of the random pixel (x, y) in the image, where $q, r \in \{0, 1\}$ and $x, y \in N+$.

$$m_{qr} = \sum_{x,y} x^q y^r I(x, y) \tag{1}$$

For example, the zero-order moment $m_{00} = \sum_{x,y} I(x, y)$ denotes the total densities of the small area around (x, y). The one-order moments are $m_{10} = \sum_{x,y} xI(x, y)$ and $m_{01} = \sum_{x,y} yI(x, y)$.

Based on Eq. (1), the corner direction is defined as the angle between the corner and the center of gravity, which is shown in Eq. (2).

$$\theta = \arctan\left(\frac{m_{01}}{m_{10}}\right) \tag{2}$$

For simplifying the next illustration, we provide the following definitions.

Definition 2 (Outer ring region). Given a pixel p, the set of pixels in a circular ring with p as the center and with the radius of 3 is defined as p's outer ring region.

Definition 3 (Middle ring region). Given a pixel p, the set of pixels in a circular ring with p as the center and with the radius of 2 is defined as p's middle ring region.

Definition 4 (Inner ring region). Given a pixel p, the set of pixels in a circular ring with p as the center and with the radius of 1 is defined as p's inner ring region.

The core of feature extraction is to obtain corners and corner directions.

3.1 Corner

The "Algorithm 1 Corner" regulates the process of obtaining corners. The algorithm takes the anticipated-testing pixel set P as the input and corner set P as the output.

Algorithm 1 *Corner*	
Input: P **Output:** P	
1:	t denotes a given minimum threshold
2:	**for all** $p \in P$ **do**
3:	$I(p)$ denotes the value of pixel p
4:	**if** ∃ a pair of pixels $\{i, i'\}$ in p's *outer* ring region are symmetrical about p and $\|I(p) - I(i)\| < t \wedge \|I(p) - I(i')\| < t$ **then**
5:	$P := P/p$
6:	$P := P/$ {the other pixels on the line between i and p}
7:	**end if**
8:	**if** ∃ a pair of pixels $\{i, i'\}$ in p's *middle* ring region are symmetrical about p and $\|I(p) - I(i)\| < t \wedge \|I(p) - I(i')\| < t$ **then**
9:	$P := P/p$
10:	$P := P/$ {the other pixels on the line between i and p}
11:	**end if**
12:	**if** ∃ a pair of pixels $\{i, i'\}$ in p's *inner* ring region are symmetrical about p and $\|I(p) - I(i)\| < t \wedge \|I(p) - I(i')\| <$ t **then**
13:	$P := P/p$
14:	**end if**
15:	**end for**
16:	**return** (P)

Algorithm 1 removes all pseudo corners from the anticipated-testing pixel set P. The specific steps are illustrated as follows:

(1) If a pair of symmetrical pixels $\{i, i'\}$ exists in p's outer ring region and $\{i, i'\}$ have similar pixel values with p, $\{i, i'\}$ are pseudo corners. All pseudo corners should be removed from P. Meanwhile, based on the adjacent pixel approximation principle [22], the other pixels on the line between i and p are also pseudo corners. Thus, we have to remove them from p (Steps 3-7).

(2) If a pair of symmetrical pixels $\{i, i'\}$ exists in p's middle ring region, and $\{i, i'\}$ have similar pixel values with p, $\{i$, i'$\}$ are pseudo corners. Meanwhile, the other pixels on the line between i and p should be removed as well (Steps 8-11).

(3) If a pair of symmetrical pixels $\{i, i'\}$ exists in p's inner ring region, and $\{i, i'\}$ have similar pixel values with p, $\{i, i'\}$ are pseudo corners (Steps 12-14).

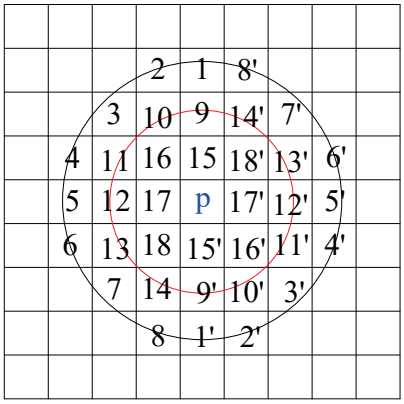


Figure 1 An example of feature extraction

Example 1. Fig. 1 shows a test regarding whether an anticipated-testing pixel p is a corner.

In Fig. 1, $\forall i \in [1, 18]$, $\{i, i'\}$ are symmetrical about p. Thus, 18 pairs of symmetrical pixels exist. p's outer ring region includes 8 pairs of symmetrical pixels—{1, 1'}, {2, 2'}, …, {8, 8'}; p's middle ring region includes 6 pairs of symmetrical pixels—{9, 9'}, …, {14, 14'}; p's inner ring region includes 4 pairs of symmetrical pixels—{15, 15'}, …, {18, 18'}.

Algorithm 1 first tests p's outer ring region. As long as any pair of pixels has similar pixel values with p, p is a pseudo corner. For example, if {1, 1'} have similar pixel values with p, p is a pseudo corner. Meanwhile, as pixels 9 and 15 are on the line between 1 and p, pixels 9 and 15 are also pseudo corners. They do not need to be tested. Therefore, we enhanced the efficiency of feature extraction by directly judging the pixels on the line between i and p without testing them.

3.2 Direction

"Algorithm 2 Direction" regulates the process of obtaining corner directions. It takes the corner set P as the input and corner direction set θ as the output. The algorithm computes the direction for each corner based on Eqs. (1) and (2).

Algorithm 2 Direction	
Input: P	
Output: θ	
1:	$\theta := \phi$
2:	**for all** p $\in P$ **do**
3:	let p's coordinate be (x, y)
4:	θ_p denotes the direction of p
5:	$m_{10} := \sum_{x,y} xI(x, y)$
6:	$m_{01} := \sum_{x,y} yI(x, y)$
7:	$\theta_p := \arctan\left(\frac{m_{01}}{m_{10}}\right)$
8:	$\theta := \theta \cup \theta_p$
9:	**end for**
10:	**return** (θ)

4 FEATURE EXTRACTION FOR MOBILE ANTI-SHAKE

Two types of mobile shakes exist in an AR system. One type is where the user consciously shakes the mobile. In this case, there is a significant change in the camera pose. The other type is where the shake happens unconsciously. In this case, the camera poses changes little. We only focus on the second case that is defined as a mobile shake herein.

This case can be attributed to three reasons:

(1) Physically unconscious shakes.
(2) Shakes caused by keyboard operation.
(3) Shakes caused by the surrounding environment.

When mobile shakes occur, the camera poses change too little to be noticed by the user. If we use the frame-by-frame tracking method [23, 24] at this time, the limited mobile computing resources will significantly affect the real-time tracking performance. Therefore, we only update the camera pose rather than recalculate it to enhance the anti-shake performance for dynamic scenarios.

"Algorithm 3 Anti-Shake" regulates the process of camera pose update. The algorithm takes the previous corner set P, previous corner direction set θ, and current anticipated-testing pixel set P' as the input. Meanwhile, the algorithm takes the new corner set P' and the updated corner direction set θ' as the output.

Algorithm 3 Anti-Shake	
Input: P, θ, P'	
Output: P', θ'	
1:	$P' :=$ Corner(P')
2:	n denotes a given positive small integer for testing mobile shake
3:	**if** $\lvert P - P' \rvert > n$ **then**
4:	add new corners into P_n
5:	**if** $\lvert P_n \rvert \neq 0$ **then**
6:	$\theta' := \phi$
7:	**for all** $p \in P \wedge \theta_p \in \theta$ **do**
8:	**if** p $\in$ P' **then**
9:	$\theta' := \theta' \cup \theta_p$
10:	**end if**
11:	**end for**
12:	$\theta' := \theta' \cup$Direction(P_n)
13:	**end if**
14:	**end if**
15:	**return** (P', θ')

When mobile shakes occur, the computing cost of obtaining the corner direction is much higher than that of obtaining a corner. To enhance anti-shake efficiency, the core idea of algorithm 3 is to only compute the new corners' directions. Therefore, we efficiently decreased the computing cost.

First, algorithm 1 is called to obtain the current corner set P' (Step 1). Next, we obtain the new corners by comparing the differences between the current corner set P' and the previous corner set P (Steps 2-4). Subsequently, if no new corner exists, ($|P_n| = 0$) (i.e., we may only lose some corners), the shake is blurred (Step 5). Thus, we omit the operation. Otherwise, if some new corners exist, the shake is clear. Thus, we need to update the corner directions (Steps 6-14). In the update process, if P and P' have the same corner p (Steps 7 and 8), p's direction θ_p does not need to be recalculated (Step 9). That is, we only need to compute a small number of directions by calling algorithm 2 (Step 12).

5 ALGORITHM COMPARISON

Our algorithm (EAS) reduces the amount of computation and storage compared to the other two algorithms (SIFT and ORB). To more clearly prove our algorithm's advantage, we compare all algorithms in terms of space and time complexity in Tab. 1. Space complexity decides the storage (memory consumption). Time complexity decides the computation (efficiency).

Table 1 Algorithm comparison

	EAS	SIFT	ORB
Space complexity	$O(n)$	$O(n^2)$	$O(n \times \log_2^n)$
Time complexity	$O(n)$	$O(n^2)$	$O\left(n \times (\log_2^n)^2\right)$

6 EXPERIMENT EVALUATION

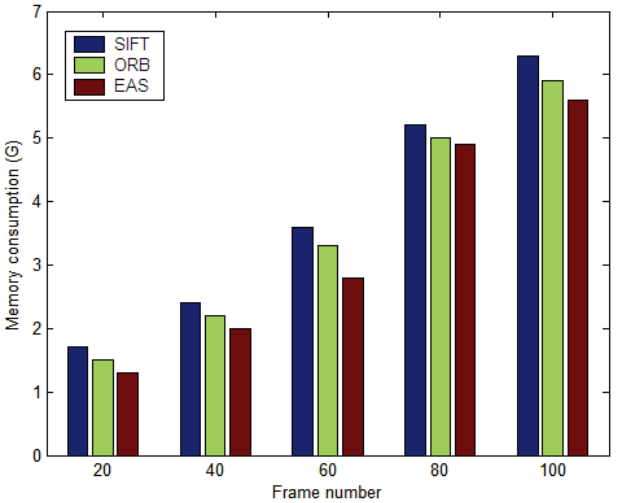
SIFT and ORB are classical methods and are related to our work. Thus, to verify the feasibility and practicality of EAS, we compare the performance of SIFT, ORB, and EAS from the aspects of memory consumption, efficiency, and accuracy.

In our experiments, we used C++ to implement the synthetic datasets. The P/S system was operating on a server with 128 GB memory, 64 KB L1 cache, 512 KB L2 cache, and CentOS 6.6 operating system. Meanwhile, we calculated the average value of 10 similar experiments to eliminate experimental errors.

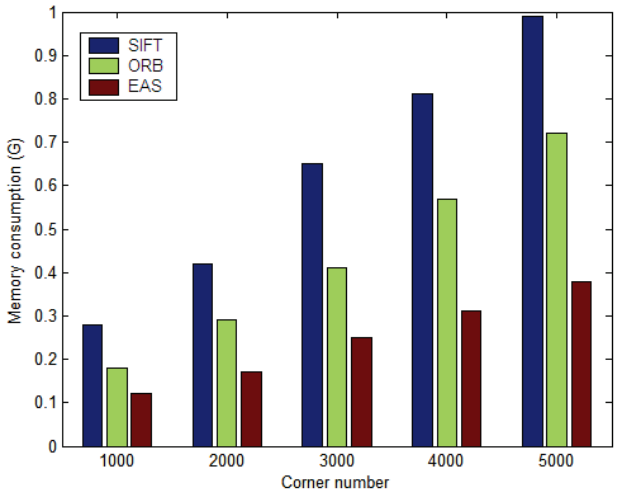
6.1 Memory Consumption

The frame number and corner number primarily affect memory consumption (RAM). The experimental results are shown in Fig. 2, where the y-axis denotes the memory consumption (units: GB). Meanwhile, the x-axis denotes the frame number and corner number separately.

As SIFT has the best judgment ability, SIFT consumes the most memory, as shown in Fig. 2. The ORB feature extraction algorithm detects the pseudo corners that require non-extreme constraints. Thus, ORB consumes more memory than EAS. The memory consumption of SIFT is almost double that of EAS and that of ORB is 1.5-fold that of EAS. Therefore, our scheme consumes less memory than SIFT and ORB during mobile shakes.



(a) Frame number



(b) Corner number

Figure 2 The comparison of memory consumption

6.2 Efficiency

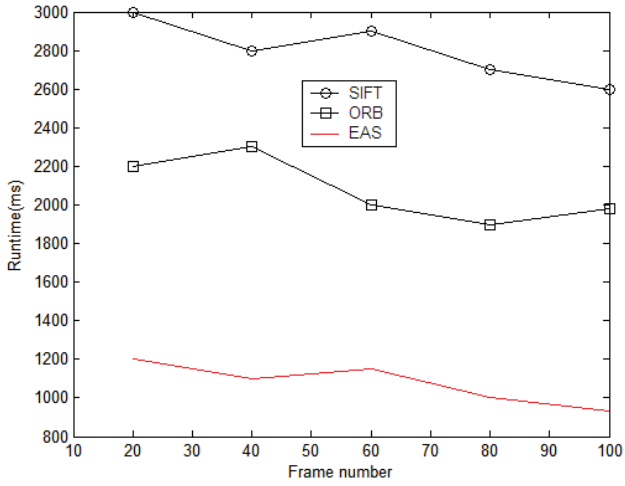
Given sufficient computational memory, our efficiency (CPU) experiment tests the runtime for real-time tracking of mobile shakes. The experiment requires the user to hold a mobile with only one hand and attempt to keep his/her hand steady.

Meanwhile, the user should aim at the tag in the scenario wherein the tag area is approximately 1/6 of the entire screen. The window size is 30 × 30 pixels. The given minimum threshold is $t = 1$. The given positive small integer is $n = 20$.

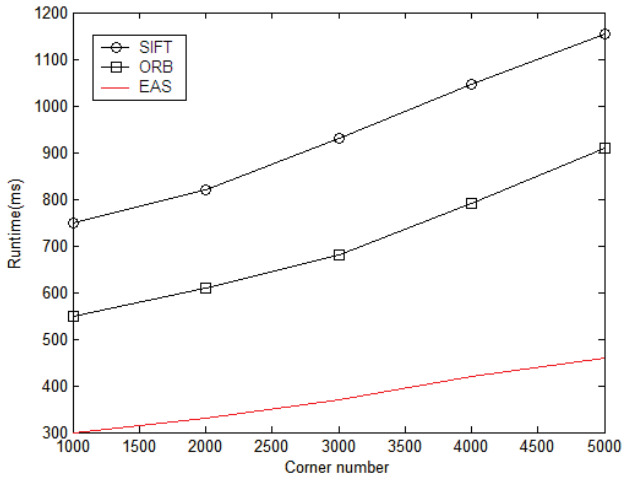
We continuously record the runtime of real-time tracking every 20 frames. The experimental results are shown in Fig. 3. The x-axis of Fig. 3 represents the frame number and corner number separately. Meanwhile, the y-axis of Fig. 3 represents runtime (units: ms).

When mobile shakes occur, SIFT does not directly test the corners. Meanwhile, SIFT tracks the camera pose frame-by-frame based on recalculations. Thus, as shown in Fig. 3, the runtime of SIFT is approximately 2.5 times more than EAS. ORB would detect much more pseudo corners than EAS during mobile shakes. Thus, the performance is

obviously lower than EAS. When mobile shakes occur, EAS directly detects the corner, only updates the camera pose without recalculation, and avoids detecting pseudo corners. The runtime of EAS is only approximately 40% that of SIFT and 55% that of ORB. Therefore, from Fig. 3, we found that EAS is more efficient than SIFT and ORB for mobile anti-shake.



(a) Frame number



(b) Corner number

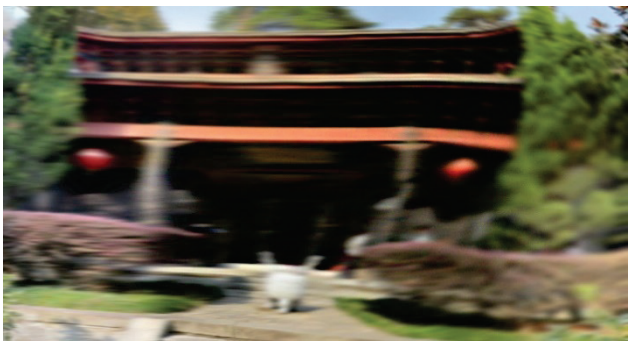
Figure 3 The comparison of efficiency

6.3 Accuracy

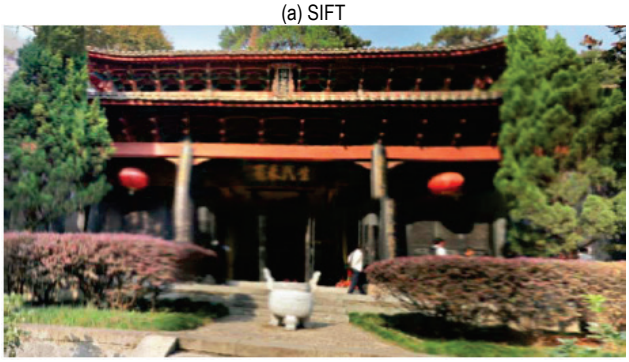
The experiment extracts features of mobile shake images in real scenarios. A set of experimental results are shown in Fig. 4.

Figs. 4(a), 3(b), and 3(c) show the accuracy of SIFT, ORB, and EAS, respectively. In Fig. 4, SIFT uses a 3 × 3 image class. The detection radius of ORB and EAS are both 3. All images are detected during mobile shakes.

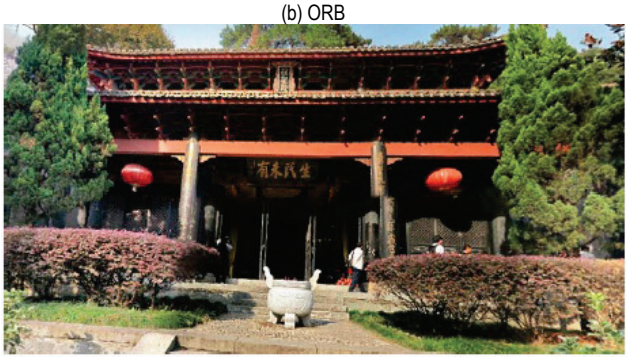
The corner detection of SIFT is not direct, and SIFT incurred much computing cost for mobile shakes. Thus, from Fig. 4, we observed that EAS's image provides more clarity than SIFT's image. As ORB detects a larger number of pseudo corners than EAS, and EAS only updates the added corners, the image accuracy of ORB is lower than that of EAS. Therefore, by Fig. 4, EAS is obviously more suitable for mobile anti-shake than SIFT and ORB.



(a) SIFT



(b) ORB



(c) EAS

Figure 4 The comparison of accuracy

7 CONCLUSION

For mobile shake application scenarios, some existing methods exhibit low efficiency and/or accuracy. To solve this issue, we proposed an efficient feature extraction scheme for mobile anti-shake in AR. The scheme directly detected corners avoiding the non-extreme constraint to enhance the efficiency of feature extraction. Meanwhile, the scheme only updated the added corners for mobile shakes to enhance the feature extraction accuracy. The memory consumption of SIFT was almost double that of EAS and of ORB was 1.5-fold that of EAS. The runtime of EAS was only approximately 40% and 55% that of SIFT and ORB, respectively. Therefore, the experimental results demonstrated that our scheme was more suitable for mobile anti-shake than most existing methods.

Acknowledgment

This research was supported by the National Natural Science Foundation of China (Nos. 61762055, 61662039 and 61462048); the Jiangxi Provincial Natural Science Foundation of China (Nos. 20122BAB211029, 20161BAB202036 and 20171BAB202004); and the Jiangxi Provincial Social Science "13th Five-Year" (2016) Planning Project of China (No. 16JY19).

8 REFERENCES

[1] Khalifa, F. A., Semary, N. A., El-Sayed, H. M., & Hadhoud, M. M. (2016). Markerless Tracking for Augmented Reality Using Different Classifiers. *The 1st International Conference on Advanced Intelligent System and Informatics*, 28-30. https://doi.org/10.1007/978-3-319-26690-9_3

[2] Cui, Z., Wu, Z., Zhou, C., Gao, G., Yu, J., Zhao, Z., & Wu, B. (2016). An Efficient Subscription Index for Publication Matching in the Cloud. *Knowledge-Based Systems, 110*, 110-120. https://doi.org/10.1016/j.knosys.2016.07.017

[3] Cui, Z., Zhu, H., Shi, J., Chi, L., & Yan, K. (2016). Efficient Authorisation Update on Cloud Data. *International Journal of Web and Grid Services, 12*(2), 109-141. https://doi.org/10.1504/IJWGS.2016.076594

[4] Lee, L. S. A., Ng, G. W., Tan, K. Y., Shaharuddin, S. S., & Wan-Busrah, S. F. (2018). Integrating Interactive Multimedia Objects in Mobile Augmented Reality for Sarawak Tourism. *Advanced Science Letters, 24*(2), 1017-1021. https://doi.org/10.1166/asl.2018.10678

[5] Erol-Kantarci, M. & Sukhmani, S. (2018). Caching and Computing at the Edge for Mobile Augmented Reality and Virtual Reality (AR/VR) in 5G. *Ad Hoc Networks*, 169-177. https://doi.org/10.1007/978-3-319-74439-1_15

[6] Paulo, M. M., Rita, P., Oliveira, T., & Moro, S. (2018). Understanding Mobile Augmented Reality Adoption in a Consumer Context. *Journal of hospitality and tourism technology*. https://doi.org/10.1108/JHTT-01-2017-0006

[7] Kourouthanassis, P. E., Boletsis, C., & Lekakos, G. (2015). Demystifying the Design of Mobile Augmented Reality Applications. *Multimedia Tools and Applications, 74*(3), 1045-1066. https://doi.org/10.1007/s11042-013-1710-7

[8] Nam, Y. (2015). Designing Interactive Narratives for Mobile Augmented Reality. *Cluster Computing, 18*(1), 309–320. https://doi.org/10.1007/s10586-014-0354-3

[9] Rister, B., Wang, G., Wu, M., & Cavallaro, J. R. (2013). A Fast and Efficient Sift Detector Using the Mobile GPU. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 26-31. https://doi.org/10.1109/ICASSP.2013.6638141

[10] Kim, G., Lee, K., Kim, Y., Park, S., Hong, I., Bong, K., & Yoo, H. J. (2015). A 1.22 Tops and 1.52 Mw/Mhz Augmented Reality Multicore Processor with Neural Network NoC for HMD Applications. *IEEE Journal of Solid-State Circuits, 50*(1), 113-124. https://doi.org/10.1109/JSSC.2014.2352303

[11] Okamoto, T., Onda, S., Yanaga, K., Suzuki, N., & Hattori, A. (2015). Clinical Application of Navigation Surgery Using Augmented Reality in the Abdominal Field. *Surgery today, 45*(4), 397-406. https://doi.org/10.1007/s00595-014-0946-9

[12] He, Y., Deng, G., Wang, Y., Wei, L., Yang, J., Li, X., & Zhang Y. (2018). Optimization of SIFT Algorithm for Fast-image Feature Extraction in Line-scanning Ophthalmoscope, *Optik, 152*, 21-28. https://doi.org/10.1016/j.ijleo.2017.09.075

[13] Huang, J., Zhou, G., Zhou, X., & Zhang, R. (2018). A New FPGA Architecture of FAST and BRIEF Algorithm for On-Board Corner Detection and Matching. *Sensors, 18*(4), 1014. https://doi.org/10.3390/s18041014

[14] Mur-Artal, R., Montiel, J., & Tardos, J. D. (2015). Orb-slam: a Versatile and Accurate Monocular Slam System. *IEEE Transactions on Robotics*, 31(5), 1147-1163. https://doi.org/10.1109/TRO.2015.2463671

[15] Yang, X., Wang, X. & Cheng, K. T. T. (2016). Ogb: A Distinctive and Efficient Feature for Mobile Augmented Reality. *International Conference on Multimedia Modeling*, 478-492. https://doi.org/10.1007/978-3-319-27671-7_40

[16] Morrison, J. G., Gálvez-López, D., & Sibley, G. (2016). Moarslam: Multiple Operator Augmented RSLAM. *Distributed Autonomous Robotic Systems*, 119-132. https://doi.org/10.1007/978-4-431-55879-8_9

[17] Du, Y., Miao, Z., & Cen, Y. (2014). Markless Augmented Reality Registration Algorithm Based on Orb. *International Conference on Signal Processing*, 19 -23. https://doi.org/10.1109/ICOSP.2014.7015197

[18] Sedaghat, A. & Mohammadi, N. (2018). Uniform Competency-based Local Feature Extraction for Remote Sensing Images. *ISPRS Journal of Photogrammetry and Remote Sensing, 135*, 142-157. https://doi.org/10.1016/j.isprsjprs.2017.11.019

[19] Çalışkan, A. & Çevik, U. (2018). An Efficient Noisy Pixels Detection Model for CT Images using Extreme Learning Machines. *Technical Gazette, 25*(3), 679-686. https://doi.org/10.17559/TV-20171220221947

[20] Lv, Z., Halawani, A., Feng, S., Réhman, S. u., & Li, H. (2015). Touch-less Interactive Augmented Reality Game on Vision-based Wearable Device. *Personal and Ubiquitous Computing, 19*(3-4), 551-567. https://doi.org/10.1007/s00779-015-0844-1

[21] Dunleavy, M. & Dede, C. (2014). Augmented Reality Teaching and Learning. *Handbook of research on educational communications and technology, 59*, 735-745. https://doi.org/10.1007/978-1-4614-3185-5_59

[22] Han, P. & Zhao, G. (2016). L-split Marker for Augmented Reality in Aircraft Assembly. *Optical Engineering, 55*(4), 043110. https://doi.org/10.1117/1.OE.55.4.043110

[23] Luo, Y. (2017). Computer Image Retracking Assessment Based on Pixel Tracking. *International Journal of Signal Processing, Image Processing and Pattern Recognition, 10*(1), 71-80. https://doi.org/10.14257/ijsip.2017.10.1.08

[24] Zhou, G., Zhang, G., Chang, C., & Ma, L. (2017). Research on Moving Target Tracking Based on Kalman Filter in Volleyball Video. *International Journal of Signal Processing, Image Processing and Pattern Recognition, 10*(1), 99-108. https://doi.org/10.14257/ijsip.2017.10.1.11

Contact information:

Lifen ZHANG, Assoc. Prof.
School of Information Science and Technology,
Jiujiang University, Jiujiang 332005, China
E-mail: 12437234@qq.com

Guangyong GAO, Prof., Ing, PhD
School of Information Science and Technology,
Jiujiang University, Jiujiang 332005, China
E-mail: gaoguangyong@163.com

Caixue ZHOU, Assoc. Prof.
School of Information Science and Technology,
Jiujiang University, Jiujiang 332005, China
E-mail: charlesjjjx@126.com

Zongmin CUI, Assoc. Prof., Ing, PhD
(Corresponding author)
School of Information Science and Technology,
Jiujiang University, Jiujiang 332005, China
E-mail: cuizm01@gmail.com

Lihua WANG, Lecturer
School of Information Science and Technology,
Jiujiang University, Jiujiang 332005, China
E-mail: 55689162@qq.com