Hybrid Visual Tracking for Augmented Books

Hyun S. Yang[1], Kyusung Cho[1], Jaemin Soh[1], Jinki Jung[1], and Junseok Lee[2]

[1] Department of Computer Science, Korea Advanced Institute of Science and Technology, 373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea
[2] Electronics and Telecommunications Research Institute, 138 Gajeongno, Yuseong-gu, Daejeon 305-700, Republic of Korea
{hsyang,qtboy,jmsoh,jk}@paradise.kaist.ac.kr, leejs@etri.re.kr

Abstract. The augmented book is the system augmenting multimedia elements onto a book to bring additional education effects or amusement. A book includes many pages and many duplicated designs so that tracking a book is quite difficult. For the augmented book, we propose the hybrid visual tracking which merges the merits of two traditional approaches: fiducial marker tracking and markerless tracking. The new method does not cause visual discomfort and can stabilizes camera pose estimation in real-time.

Keywords: hybrid visual tracking, augmented reality, augmented book.

1 Introduction

Recently, there have been a variety of approaches to raise educational achievement as well as enjoyment of books. As an example of these approaches, some systems have been developed to bring additional education effect or amusement in the augmented reality field, augmenting multimedia elements onto a book. Billinghust et al.'s *Magic Book* [1], Taketa et al.'s *Virtual Pop-up book* [2], and Cho et al.'s *e-Learning system* [3] are good examples. We will call this kind of systems augmented books.

Like other augmented reality systems, the most important problem of the augmented book is the registration between the real and virtual worlds. To address the registration problem, fiducial marker tracking or markerless tracking approaches are usually used. Fiducial marker tracking can support enough IDs and is fast, but it causes visual discomfort. On the other hand, markerless tracking does not cause visual discomfort and some real-time methods were proposed, but using this approach for tracking pages of a book is not adequate because a book includes tens or hundreds of pages and many pages are very similar, making it difficult to distinguish each page and, moreover, handle in real-time.

Applying either approach to the augmented book brings pros and cons, and pros and cons of either approach are opposite to the other one. We propose a hybrid method which can overcome the shortcomings of both approaches. That is, the new approach does not cause visual discomfort and can stabilize the camera pose estimation in real-time.

The remainder of this paper is organized as follows. In Section 2, we introduce fiducial marker tracking and markerless tracking methods and analyze their merits and

S.M. Stevens and S. Saldamarco (Eds.) ICEC 2008, LNCS 5309, pp. 161–166, 2008.

demerits. Section 3 focuses on the hybrid visual tracking, while Section 4 explains its implementation. Section 5 presents the results and Section 6 concludes this paper.

2 Related Work

In fiducial marker tracking, a fiducial marker is surrounded by a black rectangle or circle shape boundary for easy detection. ARToolkit [4], Matrix [5], and Cantag [6] are representative examples of fiducial markers. For the rectangular shape case, the camera pose is estimated using the projective relation between vertices in the scene and the world. A larger marker size is better for accurate and stable camera pose estimation, but users feel visually uncomfortable with large marker. Visual comfort and accuracy of camera pose estimation are in a tradeoff relation. In addition, almost all fiducial markers include a bit pattern representing IDs which are detected quickly and easily with binarization of the whole scene.

In markerless tracking, the key issue is the keypoint matching between scenes taken from different viewpoints. Many keypoints spread in the whole scene so that if keypoint matching is accurate, a much more accurate camera pose can be obtained compared to fiducial marker tracking. Furthermore markerless tracking does not cause visual discomfort because it does not use any fiducial markers. Famous algorithms include Mikolajczyk et al.'s method [8], SIFT [9], and SURF [10]; however these methods consume a great deal of time to create descriptors, so they are not suitable for augmented reality systems, which require real-time performance.

Recently, Lepetit el al.'s keypoint matching method [7] using randomized trees have been fast enough to perform in real-time and has been robust to various viewpoints so that it come into the spotlight of the community. Lepetit el al. transform a local image patch of each keypoint into the almost possible appearances, and train N randomized trees with those transformed patches. Williams et al. [11] suggest the modified randomized trees, which make the real-time training possible, and apply this method to the real-time SLAM. These two methods are state of the art real-time keypoint matching. However, these outstanding works are not suitable for the augmented book because they create a very large number of key points for many pages (approximately 10000 keypoints for 100 pages), and require a tremendous amount of memory (approximately 6GB for 100 pages), and consume a great deal of time for matching the large number of keypoints. Furthermore, if some similar pages exist in one book, page identification will be unstable.

3 Hybrid Visual Tracking

As stated in Section 2, fiducial markers can express abundant IDs, and markerless tracking does not cause visual discomfort and is accurate and stable for camera pose estimation. The hybrid visual tracking merges the merits from both approaches. The key idea is that page identification is performed with fiducial marker detection, and camera pose estimation is performed with randomized trees. We will call fiducial marker for page identification as page marker. Fig. 1 shows each page has the trees for keypoint matching. The page marker supports just page identification. After page

identification, the system loads the trees for the identified page, performs keypoint matching, and eliminates outliers of a matching result using RANSAC algorithm. If correct matches are enough, the system performs the pose estimation proposed in our previous work [3], otherwise the system detects a page marker because it is likely that the current page is changed.

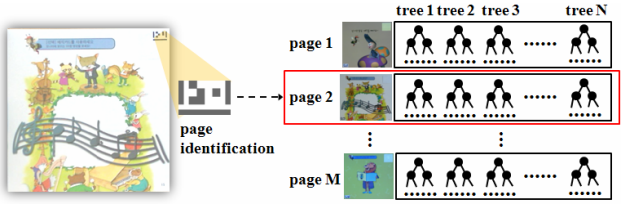

Fig. 1. The concept of Hybrid Visual Tracking

In hybrid visual tracking, a page marker is just used for page identification not for camera pose estimation so that we can reduce the size of a marker and don't need to design the shape as rectangle or circle. Therefore, a page marker can be designed in various ways not giving visual discomfort. Fig. 2 shows that our page marker is very small compared with a traditional fiducial marker. Our marker size is only 2cm x 1.3cm. In addition, randomized trees support only one page and the system loads the trees for only the current page, so we can optimize the memory space and achieve fast matching and high correct matching ratio. Compared to fiducial marker tracking, keypoints spread widely, so more accurate and stable camera pose can be estimated.

Fig. 2. *(left)* a page with a traditional fiducial marker, *(right)* a page with our page marker

4 Implementation

Some requirements are needed for the hybrid visual tracking not to cause visual discomfort and to achieve real-time performance. First of all, the size of page marker is small enough but it must not cause a little false identification. Second, for real-time performance, the keypoint detection must be performed fast enough, and marker detection must not cause overhead. Finally, an efficient data structure for randomized trees is considered to load fast them into memory once a page is changed. We will explain the keypoint detection process, the page marker detection process, and the data structure for randomized trees with considering the above requirements.

4.1 Keypoint Detection

We use the FAST detector [12] which is noticed that it takes 2 ms to find keypoints in a 640 x 480 image in real-time. This algorithm checks whether or not each pixel p is a keypoint. It tests intensities of 16 circular points whose distance is 3 from pixel p. If the absolute intensity differences between p and more than 10 contiguous circular points are all above the threshold value t, p is selected as a keypoint. The magnitude of a keypoint is the sum of the intensity difference between p and all the circular points. This algorithm is very fast because most pixels are rejected by considering only four points instead of all the circular points. We extract 200 keypoints per page for the training step and 150 for the test step using the FAST detector.

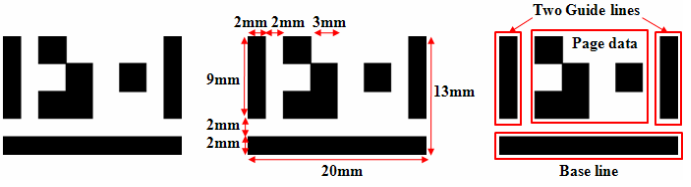

Fig. 3. Page marker design

4.2 Page Marker Design and Detection

For page marker detection, we want to use the result of the keypoint detection to reduce the candidate regions, rather than searching through the entire scene. The FAST detector is sensitive to spiky points and points that have very different intensities from their neighbors. Therefore, we design page markers like in Fig. 3, which includes many spiky points and use just black and white colors. A page marker are surrounded with the base line and the two guide lines, and includes the 12 bits pattern. To reduce the false positive ratio, we divide 12 bits into 8 bits and 4 bits which represent the ID itself and the CRC (Cyclical Redundancy Check) code of the ID, respectively, and also reject the patterns including too many white cells or too many black cells so that only the patterns with between 3 and 9 black cells are used. Finally, the page marker can distinguish 245 IDs without false detection.

For page marker detection, we select the top 50 keypoints based on magnitude and group them in a 30 pixel distance. These groups are candidate regions for a page marker. For each candidate region, CCL algorithm and PCA analysis are performed to find one base line and two guide lines. After detection, page data is decoded and an ID and a CRC code are extracted. If the CRC code is correct, then the ID is recognized as the page ID. This algorithm is very fast because it does not search through the entire scene, but only considers the keypoints detected by the FAST detector.

4.3 The Data Structure for Randomized Trees

When the original randomized trees are loaded into memory from file, tree structures are built one by one, is time-consuming. On the other hand, Williams et al.'s randomized tree can be represented as an array with only d elements instead of a tree with 2^d-1 internal nodes where d is the depth of a tree. By using the latter one, we can make it

possible to load randomized trees into memory immediately without building a tree structure.

In practice, we train 40 randomized trees with depth 10 for one page, the total amount of space for one page is 31MB, and it takes 34 ms to load them. As a result, the average correct matching rate is 70%.

5 Results

Our experiments were conducted on a 3.0GHz PC and used a book which has 30 pages. We put page markers only on odd pages and trained randomized trees. In the book, the designs of 5 and 11, 13 and 25, 15 and 27, and 17 and 23 page are exactly the same, which make it very hard to identify the pages with only randomized trees.

The hybrid visual tracking method identified pages very well and estimated camera pose robustly in difficult situations such as viewpoint changes and serious occlusion like Fig. 4. We need only 31MB space of memory to operate the system because we need only one page of data at a time.

Table. 1 shows the average amount of time needed for a 2500 frame video clip for 15 pages. As expected, the page marker detection takes so little time that it does not have an effect on performance. After loading the tree data, only 22.49ms was needed to finish the pose estimation process. That means the proposed method can achieve 44.46 fps ideally.

Fig. 4. Pose estimation under *(left and middle)* viewpoint changes, *(right)* serious occlusion

Table 1. The time needed for hybrid visual tracking

Keypoint Detection	Page Marker Detection	Keypoint Matching	RANSAC	Pose Estimation
6.05 ms	1.11 ms	2.87 ms	6.45 ms	7.12 ms

6 Conclusion

We propose the hybrid visual tracking which can efficiently estimate camera poses when there are many objects to be augmented and many duplicated designs like the augmented book. With only wide-baseline matching for each frame, it takes 22.49ms, but if the small-baseline matching technique and camera tracking techniques such as the extended kalman filter or the particle filter are considered, faster and more robust

results can be obtained for camera pose estimation. Furthermore, memory cache and background processing techniques can be used for loading randomized trees to accelerate the page loading speed.

Acknowledgments. This research was supported by the Development of Elemental Technology for Promoting Digital Textbooks and u-Learning, which is sponsored by the Ministry of Information and Communication, and the Ubiquitous Autonomic Computing and Network Project of the 21st Century Frontier R&D Program, which is sponsored by the Ministry of Information and Communication.

References

1. Billinghurst, M., Kato, H., Poupyrev, I.: The MagicBook: A Transitional AR Interface. Computers and Graphics, pp. 745–753 (2001)
2. Taketa, N., Hayashi, K., Kato, H., Noshida, S.: Virtual Pop-Up Book Based on Augmented Reality. In: Smith, M.J., Salvendy, G. (eds.) HCII 2007. LNCS, vol. 4558, pp. 475–484. Springer, Heidelberg (2007)
3. Cho, K.S., Lee, J.H., Lee, J.S., Yang, H.S.: A Realistic e-Learning System based on Mixed Reality. In: 13th International Conference on Virtual Systems and Multimedia (2007)
4. Billinghurst, M., Kato, H.: Collaborative mixed reality. In: 1st International Symposium on Mixed Reality, pp. 261–284 (1999)
5. Rakimoto, J.: Matrix: A realtime object identification and registration method for augmented reality. In: Asia Pacific Computer Human Interaction, pp. 63–68 (1998)
6. Rice, A., Beresford, A., Harle, R.: Cantag: an open source software toolkit for designing and deploying marker-based vision systems. In: 4th IEEE International Conference on Pervasive Computing and Communications (2006)
7. Lepetit, V., Fua, P.: Keypoint Recognition using Randomized trees. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(9), 1465–1479 (2006)
8. Mikolajczyk, K., Schmid, C.: An Affine Invariant Interest Point Detector. In: 5th European Conference on Computer Vision, pp. 414–431 (2002)
9. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 20(2), 91–110 (2004)
10. Bay, H., Tuytelaars, T., VanGool, L.: SURF: Speeded Up Robust Features. In: 9th European Conference on Computer Vision, pp. 404–417 (2006)
11. Williams, B., Klein, G., Reid, I.: Real-Time SLAM Relocalisation. In: 11th IEEE International Conference on Computer Vision (2007)
12. Rosten, E., Drummond, T.: Fusing points and lines for high performance tracking. In: 9th IEEE International Conference on Computer Vision, pp. 1508–1511 (2005)