Multiple Page Recognition and Tracking for Augmented Books

Kyusung Cho, Jaesang Yoo, Jinki Jung, Hyun S. Yang

Multiple Page Recognition and Tracking for Augmented Books

Kyusung Cho[1], Jaesang Yoo[1], Jinki Jung[1], and Hyun S. Yang[1]

[1] Department of Computer Science, Korea Advanced Institute of Science and Technology,
373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea
{qtboy, jsyoo, jk, hsyang}@paradise.kaist.ac.kr

Abstract. An augmented book is an application that augments virtual 3D objects to a real book via AR technology. For augmented books, some markerless methods have been proposed so far. However, they can only recognize one page at a time. This leads to restrictions on the utilization of augmented books. In this paper, we present a novel markerless tracking method capable of recognizing and tracking multiple pages in real-time. The proposed method builds on our previous work using the generic randomized forest (GRF). The previous work finds out one page in the entire image using the GRF, whereas the proposed method detects multiple pages by dividing an image into subregions, applying the GRF to each subregion and discovering spatial locality from the GRF results.

Keywords: Augmented Books, Multiple Page Tracking, Markerless Visual Tracking, Augmented Reality

1 Introduction

Recently, there have been a variety of approaches to enhance books by adding digital information. As an example of these approaches, some applications have enhanced real books by means of augmentation with 3D virtual objects using augmented reality technology. We refer to these applications as the 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, most of the augmented book applications employ fiducial markers because they are easy to recognize and track [1],[2],[3]. However, fiducial markers can lead to visual discomfort due to their distinct shapes, so there is a growing tendency to employ markerless page tracking methods [4], [5], [6], [7]. Especially, Cho et al.'s work [7] has presented the markerless tracking method capable of handling a lot of pages (about 200 pages) using the generic randomized forest (GRF).

All proposed markerless tracking methods for augmented books have not so far recognized multiple pages simultaneously. This imposes restrictions on book designs and users' activitie. If multiple sheets of paper can be recognized, more various interactions on an augmented book may be possible. For multiple page recognition and tracking, we propose the novel markerless tracking method which extends our previous work [7]. The previous work finds out one page in the entire image using the GRF, whereas the proposed method detects multiple pages by dividing an image into subregions, applying the GRF to each subregion and discovering spatial locality from the GRF results.

The remainder of this paper is organized as follows. In Section 2, we review the GRF in the previous work. Section 3 focuses on multiple page recognition and tracking of the proposed method. Section 4 presents the experiment results and Section 5 concludes this paper.

2 Generic Randomized Forest

Given an input image by a camera, an augmented book application needs to recognize a page and perform wide-baseline keypoint matching for calculating its initial pose. For this purpose, in our previous work [7], the generic randomized forest (GRF) which extends the original randomized forest (RF) proposed in [8] is presented. An original RF is used to match keypoints against already-trained keypoints coming from one object. It consists of several randomized trees where internal nodes test the intensity difference between two pixels of an image patch around a keypoint and leaf nodes store the probability distribution related to all keypoints.

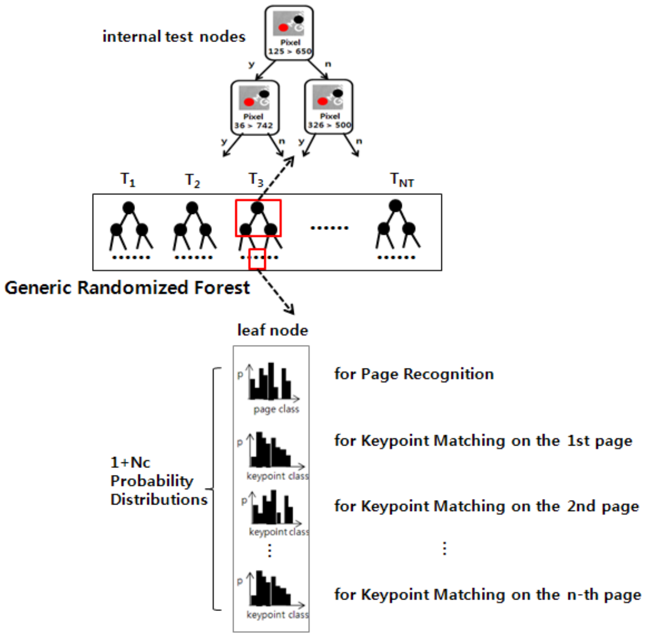

Fig. 1. Generic Randomized Forest

To handle a lot of pages, the GRF stores the probability distribution for page recognition as well as the probability distribution for keypoint matching of each page in leaf nodes like in Fig. 1. If a book has *Nc* pages, each leaf node of the GRF includes *Nc+1* probability distributions. In real-time, *N* keypoints are extracted from an image by any detector and then passed through the GRF to recognize a current page, as in (1).

$$\begin{aligned} Page\ \hat{\imath} &= argmax_i\ P(C = i|T_1, \cdots, T_{NT}, m_1, \cdots, m_N) \\ &= \underset{i}{argmax} \frac{1}{N} \sum_{j=1}^{N} \frac{1}{NT} \sum_{t=1}^{NT} P(C = i|leaf(T_t, m_j)) \end{aligned} \tag{1}$$

,where *leaf(T_t,m_j)* is the leaf node which the i-th keypoint m_i reaches in the t-th tree T_t. If the i-th page is recognized as a result of (1), then keypoint matching for the i-th page considers only the i-th probability distribution stored in the leaf nodes. Keypoint m_j is matched, as in (2).

$$\begin{aligned} Keypoint\ \hat{k} &= P\left(K = k\middle|T_1, \cdots, T_{NT}, m_j\right) \\ &= \underset{k}{argmax} \frac{1}{NT} \sum_{t=1}^{NT} P(K = k|leaf(T_t, m_j)) \end{aligned} \tag{2}$$

3. Proposed Method

In the following we describe our proposed method capable of recognizing and tracking multiple pages, which is intended for obtaining page information including IDs and pose information of all pages visible in an image captured by a camera. The final page information (**P**) is a set of visible pages' information *p = (ID, R, T)*, where *R* is a 3x3 rotation matrix and *T* is a 3-translation vector representing a page pose.

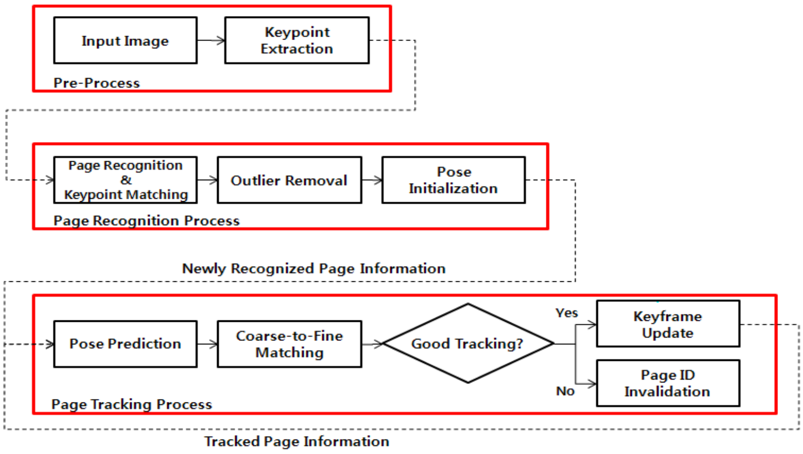

Fig. 2. Overview of the proposed method

Prior to the real-time processing, an offline training process is required. In the training process, we take one picture of each page included in the book, extract the keypoints, and train the GRF in such a way as to be explained in [7].

The proposed method consists of three main processes: pre-process, page recognition process, and page tracking process as shown in Fig. 2. In real-time, keypoints are first extracted from the input image by the FAST detector [9] because it is well known to be very fast. However, keypoints extracted by the FAST detector do not contain scale information, so we build three levels of an image pyramid and extract keypoints from each level. They are conveyed to the page recognition process to recognize pages appearing newly in the image except already-tracked pages. For the page tracking process, we apply the coarse-to-fine matching technique as well as the adaptive keyframe-based tracking which updates the keyframe adequate for tracking as time goes by. The tracking process is motivated by Klein et al.'s work [12] which is well known as PTAM.

We explain details of the page recognition process in Section 3.1 and the page tracking process in Section 3.2.

3.1 Page Recognition Process

In the previous work, we assumed that a single page at most among the whole pages is visible. Therefore, all keypoints extracted from the entire image pass through the GRF and the page can be recognized using (1). However, the assumption is invalid in this work because our target pages in the image may be more than one. Our strategy to deal with this problem is to divide the image into subregions, inspect which pages are included in each subregion, and put together results using spatial locality. The left image of Fig. 3 shows the original image including two pages of which page IDs are 4 and 7 and the right image shows m x n subregions overlaid with keypoints.

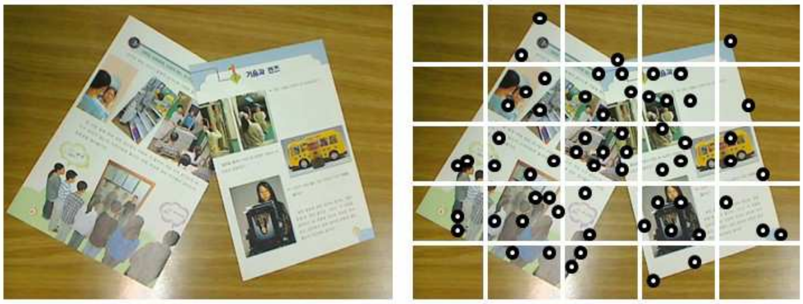

Fig. 3. (left) Original image, (right) Subregions overlaid with keypoints

We apply the GRF to each subregion individually in a way that only keypoints belonging to each subregion participate in calculating (1). Equation (1) is originally intended for averaging the probability distributions with respect to all keypoints and selecting the page with the highest probability, whereas we consider the top *T* pages instead of the only top one page. $Rank_r(i)$ is referred to as the ranking of the i-th page in the average probability distribution of the r-th subregion. To evaluate the likelihood that the r-th subregion might contain the i-th page, *Effect_r(i)* is defined as (3).

$$Effect_r(i) = \begin{cases} \mathrm{T} + 1 - Rank_r(i) & \text{if } Rank_r(i) \le \mathrm{T} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

If *$Effect_r(i)$* is greater than 0, we call the relationship between the r-th subregion and the i-th page "*the r-th subregion is effective to the i-th page*". Fig. 4 (a) shows the page IDs of which *$Effect_r(i)$* are greater than 0 in each region in decreasing order when T is 3. As shown in Fig. 4. (a), if the page ID actually comes from the true page, subregions with the same page ID have a tendency to gather together. We refer to this property as the spatial locality. Thus, if a large connected subregion group with the same ID is detected, the group is likely to include the page with the ID. *LCR(i)* is referred to as a set of 4-connected effective subregions related to i-th page. To evaluate the likelihood that the entire image might contain the i-th page, *Score(i)* is defined as (4).

$$Score(i) = \sum_{\mathrm{r} \in \mathrm{LCR(i)}} Effect_r(i) \tag{4}$$

Fig. 4. (b) and (c) show the top two connected subregion groups with the highest score, of which page IDs are 7 and 4. As shown in the both figures, it seems that the connected subregion group segments the real page region roughly.

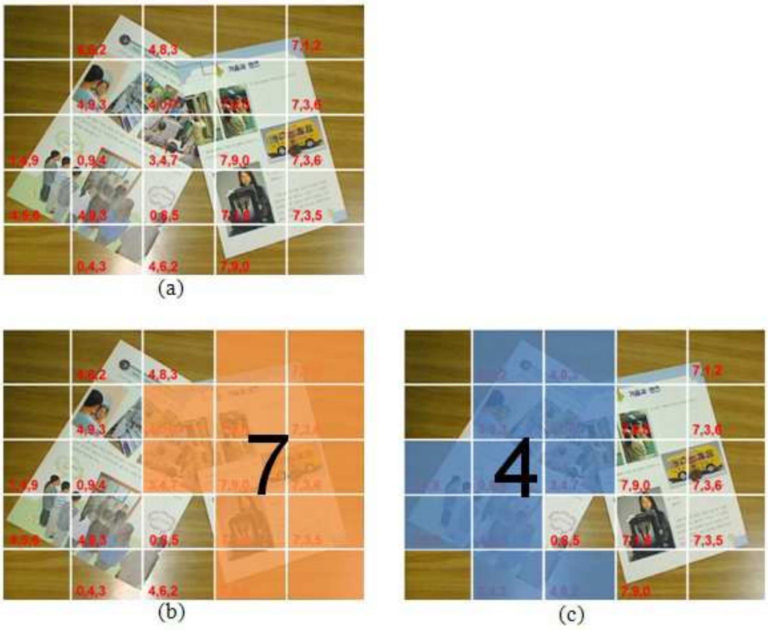

Fig. 4. Spatial locality of subregions

If pages which have already been recognized and tracked exist, it is not required to reconsider them in the page recognition process, so they are skipped. For all the newly recognized pages, keypoint matching is performed as in the previous work to calculate their initial poses. If the 7th page is recognized, only the keypoints belonging to *LCR(7)* participate in the keypoint matching step using (2). There might exist outliers among the keypoint matching result, so it is required to remove the outliers. We employ the PROSAC method [10] for that purpose. Even if the page recognition results in wrong page IDs, the outlier removal step can filter them out. As the last step of the page recognition process, initial poses of recognized pages are calculated from the matching result by using [11]. The final page information *P*, a set of visible pages' information *p=(ID, R, t)*, is conveyed to the page tracking process and then poses are refined.

3.2 Page Tracking Process

When the new page is recognized by the page recognition process, we initially use the training image of the page as a keyframe to track the page. The pose of the page belonging to the training image has already been calculated in the training process. However, this is quite dangerous in terms of tracking stability because the training image could be taken from a different camera in different environments. Thus, we update the current frame as a keyframe just in case that it describes the page better than the existing keyframe according to the update score after every tracking success.

At every frame the following procedure is performed for tracking each page.

1) A prior pose is estimated from a motion model.
2) Map points in the world are projected into the image according to the estimated prior pose in step 1.
3) A coarse search is performed with 60 map points and the camera pose is refined.
4) A fine search is performed with at most 500 map points and the final pose is computed from the matching.
5) Update the motion model.

Camera motion M can be parameterized with a six-vector μ, which consists of three elements for translation and the remainder for rotation, using the exponential map [13]. Thus, given a camera pose P which transforms a point in a world coordinate into a point in a camera coordinate, the new camera pose $\hat{P}$ can be estimated as in (5) [12].

$$\widehat{\mathrm{P}} = \mathrm{M}\,\mathrm{P} = \exp(\mu)\,\mathrm{P} \tag{5}$$

,where $P = [R\ t]$ and R and t are the camera rotation matrix and translation vector, respectively. The decaying velocity motion model is used which slows and stops eventually in case of the lack of new measurements.

To find matching pairs between map points in the world coordinate system and keypoints in a current image frame, a map point (***X***) is projected into an image, as in (6).

$$x = K[R\ t]\boldsymbol{X} = K[R\ t]\begin{bmatrix} X \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{bmatrix} \quad (6)$$

,where x is a 2D point in an image coordinate and K is the intrinsic matrix of the camera.

We perform an affine warping to approximate viewpoint changes between the 8x8 image patch generated from the keyframe of the world map and the current camera position, as described in [12]. The determinant of the affine warping matrix is used to determine the pyramid level at which the patch could be searched. The best match between the projected map point and a keypoint in the current image frame can be found within a fixed radius around the projected map point position by evaluating zero-mean SSD scores within the circular search region.

To make the page tracking process more robust to rapid camera motions, patch search and pose update are done twice. First, a coarse search is done with only 60 map points from the highest levels of the image pyramid of the current frame. Patch search is performed with a larger radius and a pose is refined with the successful matching pairs, by minimizing the Tukey biweight objective function [14] of the reprojection error iteratively. With the refined pose, a fine search is done up to 500 map points. Now the patch search is performed with a smaller search region. The final camera pose is calculated eventually and the camera motion is updated from the difference between the initial and final camera pose of the frame.

A tracking would likely fail by a motion blur, occlusion, or an incorrect position estimate. Thus, if a fraction of the keypoint matching falls below a certain threshold, it is considered as a tracking failure and this page is eliminated from the tracked page list, so the page can be re-recognized through the page recognition process.

The tracking quality in the page tracking process depends on the quality of the keyframe because it is used in the patch search. However, because the initial keyframe of each page is from the offline stage and it could be captured from a different camera in different environments, the fraction of the keypoint matching is likely to fall, which might cause the unexpected tracking failure as well as a poor tracking quality. Thus, the goal of the keyframe update is to capture the image frame as a keyframe for the page which describes the page well, satisfying the following three conditions

1) An image is clear enough with no motion blur.
2) The area of a page appears as much as possible in the image and it is captured as large as possible in the image.
3) The page plane and camera center are orthogonal.

The total score function of the t-th frame is the weighted sum of the three sub score functions of $\mathrm{Score}_{\mathrm{ZMSSD}}$, $\mathrm{Score}_{\mathrm{area}}$, and $\mathrm{Score}_{\mathrm{ortho}}$, as shown in (7)

$$\begin{aligned} Score_{total}\,(I_t) = {} & \omega_1\, Score_{ZMSSD}\,(I_t) \\ & + \omega_2\, Score_{area}\,(I_t) \\ & + \omega_3\, Score_{ortho}(I_t) \quad (7) \end{aligned}$$

,where $\text{Score}_{\text{ZMSSD}}$, $\text{Score}_{\text{area}}$, and $\text{Score}_{\text{ortho}}$ represent the above conditions in sequence and ω_1, ω_2, and ω_3 are the weight factors of the three scores which represent their importance, respectively. $\text{Score}_{\text{ZMSSD}}$ measures how similar the adjacent frames are with no motion blur as in (8). ZMSSD is the zero-mean squared sum of distance between two adjacent blurred images at frame t (BI_t) and t-$1(BI_{t-1})$.

$$Score_{ZMSSD}\,(I_t) = 1 - \frac{ZMSSD\,(BI_t,BI_{t-1})}{ZMSSD\_max} \tag{8}$$

$\text{Score}_{\text{area}}$ measures how much portion of the pages are shown within the image at frame t, as in (9).

$$Score_{area}\,(I_t) = \frac{Area_{\,t}}{ImageSize}\;\frac{Area_{\,t}}{AreaOfPage_{\;t}} \tag{9}$$

,where Area_t is the area of the page shown in the image at frame t in pixel scale and $\text{AreaOfPage}_\text{t}$ is the area of the page, including the area beyond the image boundary after projecting four boundary points of the page into the image according to the camera pose.

$\text{Score}_{\text{ortho}}$ measures how orthogonal the page is to the camera z vector, by calculating an inner product of the camera z vector $(\text{CamZ}_{\text{z,t}})$ and the inverse of the page normal vector $(-\text{PageNorm}_{\text{z,t}})$, as in (10).

$$Score_{ortho}(I_t) = CamZ_{z,t} \cdot (-PageNorm_{z,t}) \tag{10}$$

Therefore, the higher $\text{Score}_{\text{total}}$ is the more accurately the page tracking process tracks the pose of a page. Fig. 5. shows the progress of the keyframe update from (a) to (f), where (a) is the training image used as the initial keyframe; it is quite different from the other keyframes captured in real-time. Progressing from (b) to (f), it is ascertained that the keyframe is more orthogonal and fit to the image to be favorable for tracking.

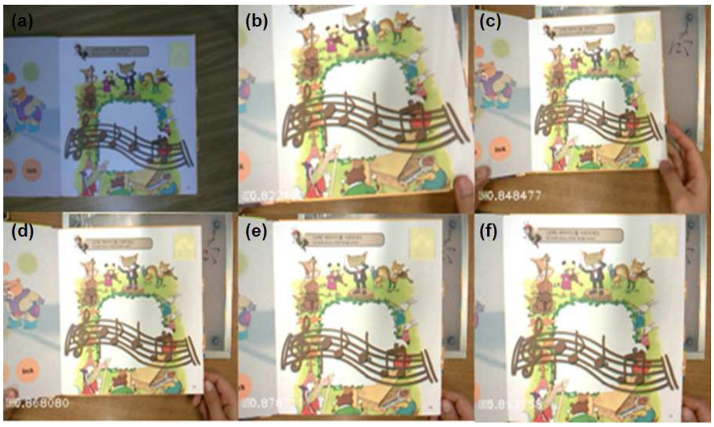

Fig. 5. The progress of the keyframe update from (a) to (f)

4. Experiment Results

For the experiment, a desktop of 3.17GHz Core 2 duo CPU with 2GB memory and a NVIDIA GeForce 8600 graphic card were used. A logitech Q9000 webcam was attached to the desktop. A 640x480 image was obtained from the camera. Although the dual core CPU was used in the experiment, the proposed method can be performed on a single core CPU because it uses a single thread.

The experiments were performed with a science textbook including 166 pages for elementary students. We took one picture of each page and extracted 250 keypoints per page. The keypoints were used in training the GRF with the number of trees *NT* = 40 and a depth of *d* = 10.

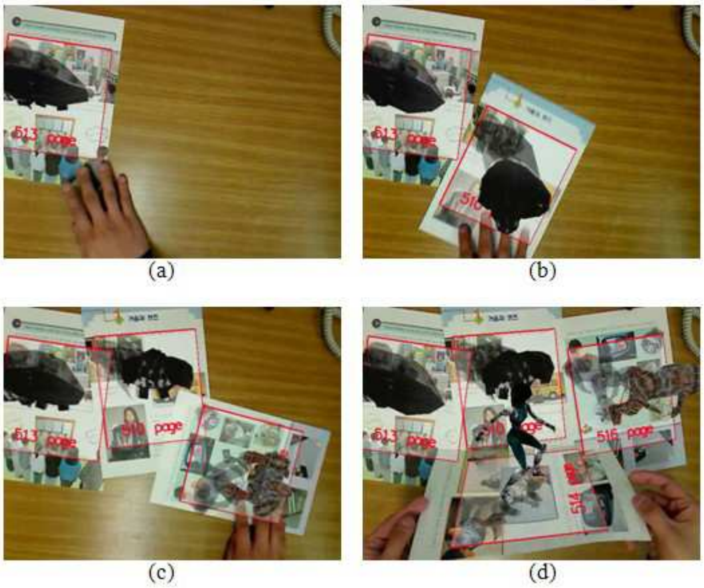

Fig. 6. Multiple Page Recognition and Tracking

In real-time, 300 keypoints are extracted from an image captured by the camera, the image is divided into 4 x 5 subregions and keypoints belonging to each subregion are passed through the GRF. We consider the top 10 page IDs in each subregion (T=10 in Equation (3)). Although the proposed method can recognize and track regardless of the number of visible pages, we impose a limitation to handle up to four pages due to the real-time constraint (more than 25 fps). Fig 6. (a)-(d) show that one to four pages are recognized and tracked.

Besides the ordinary conditions such as Fig. 6., multiple pages are tracked well in situations with dramatic viewpoint variation, scale variations, illumination variations, and partial occlusions thanks to the adaptive keyframe-based tracking, as shown in Fig. 7.

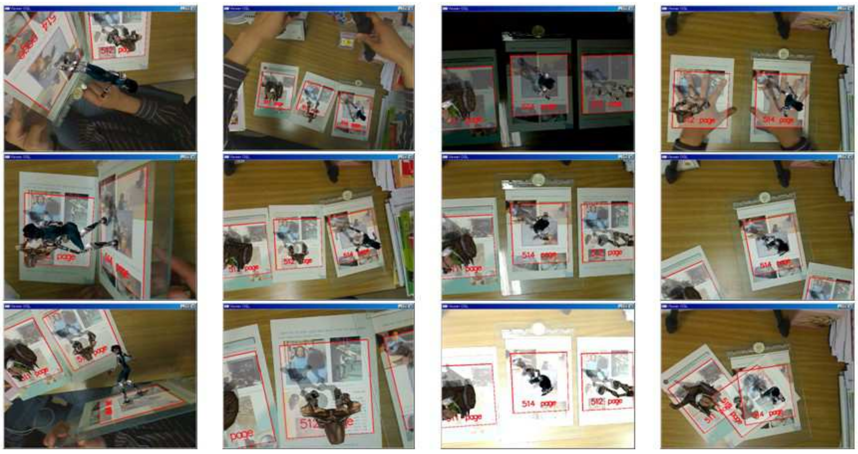

Fig. 7. Tracking in situations with dramatic viewpoint variation (1st column), scale variations (2nd column), illumination variations (3rd column), and partial occlusions (4th column)

One of the most important things to decide a tracking quality is the jitter. To measure the jitter, we used the reprojection error as measurement and compared the adaptive keyframe based tracking with the fixed keyframe based tracking, where the training image was always used as the keyframe. Fig. 8. (a) shows a setup for measuring the jitter, where a test page and the camera did not move and 1000 frames were selected to measure the reprojection error.

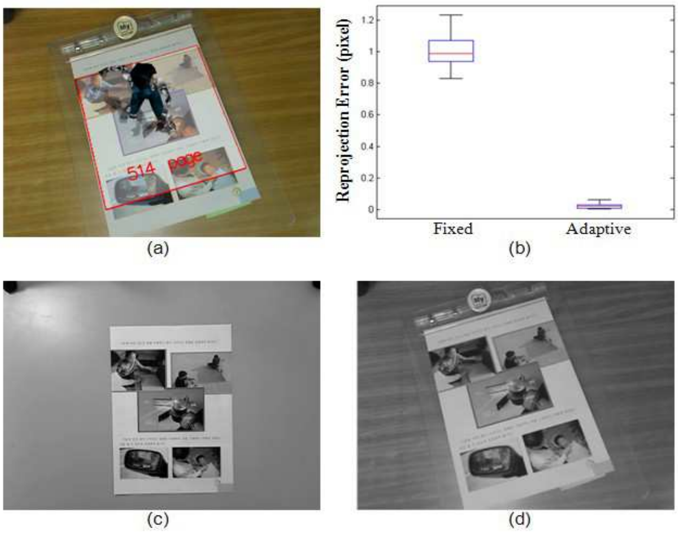

Fig. 8. (a) a setup for measuring the jitter, (b) the comparison result of the fixed and the adaptive methods, and the keyframes (c) for the fixed and (d) for the adaptive methods

Fig. 8. (b) shows the comparison result using a boxplot. The means of the fixed and the adaptive keyframe based tracking are 1.002 pixels and 0.026 pixels, respectively. This indicates that the adaptive method results in much less jitter than the fixed method. Fig. 8. (c) and (d) show the keyframes for the fixed and the adaptive methods.

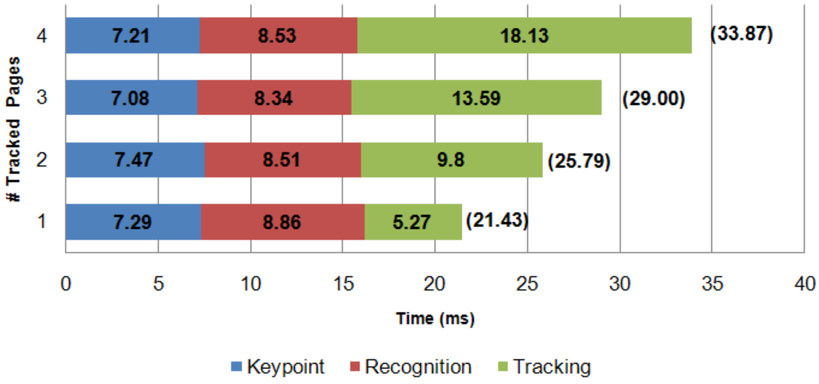

Fig. 9. The required times

Finally, we observed the required time as the number of tracked pages increases. Fig. 9. shows the average times for tracking in consecutive 100 frames according to the number of tracked pages. As expected, the more pages are tracked, the time is longer. The overall time consists of the keypoint extraction time, the recognition time, and the tracking time. The number of tracked pages has no effect on the times required for the keypoint extraction and recognition, but the tracking time increases linearly. If the number of pages tracked at a time is restricted to four, the proposed method can achieve approximately 30 fps.

5. Conclusion

For augmented books, this paper presents the markerless visual tracking method capable of recognizing and tracking multiple pages in real-time. The proposed method builds on our previous work using the generic randomized forest (GRF). The previous work finds out one page in the entire image using the GRF, whereas the proposed method detects multiple pages by dividing an image into subregions, applying the GRF to each subregion and discovering spatial locality from the GRF. We also propose the adaptive keyframe-based tracking which updates the keyframe adequate for tracking as time goes by.

As a result, the proposed method is robust to various situations with dramatic viewpoint variation, scale variations, illumination variations, and partial occlusions. Thanks to the adaptive keyframe-based tracking, the jitter can be much reduced. Although the required time increases linearly as the number of tracked pages increases, our method can achieve 30 fps if we impose a limitation to handle up to four pages.

In the future, we have a plan to perform more experiments related to the accuracy of page recognition and a page pose. In addition, our method requires the complementary relationship between the recognition and tracking processes. We expect that this makes our method more efficient and faster.

References

1. Billinghurst, M., Kato, H., Poupyrev, I.: The MagicBook: A Transitional AR Interface, Computers and Graphics, pp. 745-753 (2001)
2. 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)
3. Fumihisa S., Yusuke Y., Koki F., Toshio S., Kenji K., Asako K., Hideyuki T.: Vivid Encyclopedia: MR Pictorial Book of Insects. In Proc. Virtual Reality Society of Japan Annual Conference (2004)
4. Taketa N., Hayash K., Kato H., Nishida S.: Virtual pop-up book based on augmented reality. In Proc. HCI (2007), pp.475–484.
5. Scherrer C., Pilet V., Fua P., Lepetit V.: The haunted book. In: 7th IEEE/ACM International Symposium on Mixed and Augmented Reality (2008)
6. Kim. K., Park J., Woo W.: Marker-Less Tracking for Multi-layer Authoring in AR Books, In: 8th International Conference on Entertainment Computing, pp.48-59 (2009)
7. Cho, K., Yoo, J., Yang H.S.: Markerless Visual Tracking for Augmented Books, In: Joint Virtual Reality Conference EGVE-ICAT-EURO VR, pp. 13-20 (2009)
8. Lepetit V., Fua P.: Keypoint Recognition using Randomized trees. IEEE Transactions on Pattern Analysis and Machine Intelligence (2006), vol.28, no.9, pp.1465-1479.
9. Rosten E., Drummond T.: Machine learning for high-speed corner detection. In: 9th European Conference on Computer Vision (2006), pp.430-443.
10. Chum O., Matas J.: Matching with PROSAC-Progressive Sample Consensus. In: IEEE Conference on Computer Vision and Pattern Recognition (2005), pp.220-226.
11. Klein G.: Visual Tracking for Augmented Reality, In: PhD. Thesis, University of Cambridge (2006), pp.165-169.
12. Klein G., Murray D.: Parallel tracking and mapping for small AR workspaces. 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (2007), pp.255-234.
13. Varadarajan V.: Lie Groups, Lie Algebras and Their Representation. SpringerVerlag, 1974.
14. Huber P.: Robust Statistics. Wiley, 1981.