Joint Virtual Reality Conference of EGVE - ICAT - EuroVR (2009)
M. Hirose, D. Schmalstieg, C. A. Wingrave, and K. Nishimura (Editors)

Markerless Visual Tracking for Augmented Books

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

[1]Korea Advanced Institute of Science and Technology, Korea

Abstract

An augmented book is an application that augments such multimedia elements as virtual 3D objects, movie clips, or sound clips to a real book using AR technologies. It is intended to bring additional education effects or amusement to users. For augmented books, this paper presents a markerless visual tracking method which recognizes the current page among numerous pages and estimates its 6 DOF pose in real-time. Given an input image by a camera, the tracking method first recognizes a page and performs wide-baseline keypoint matching at the same time. For that purpose,a generic randomized forest (GRF) is proposed which extends the randomized forest (RF) proposed by Lepetit et al. which only performs wide-baseline keypoint matching. The proposed GRF is capable of simultaneous page recognition and wide-baseline keypoint matching. Once a page is recognized, the tracking method executes the page tracking process without page recognition until the page is turned. The page tracking process selects a keyframe of the page adequate for tracking and employs a coarse-to-fine approach. As a result, the tracking method shows robustness to viewpoint and illumination variations and performance of more than 30 fps for augmented books.

Categories and Subject Descriptors (according to ACM CCS): H.5.1 [Multimedia Information Systems]: Artificial, augmented, and virtual realities—Augmented Book, Markerless Visual Tracking

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 via augmented reality technology. These applications are referred to here as augmented books. An augmented book can raise users' understanding of the contents of the book and provide visual impressions for users. This makes it popular for educational [CLY07], [FYK*04], [THKN07], entertainment [BKP01], [JRP*05], art [SPFL08] and advertisement applications.

Like other augmented reality systems, the most important problem with an augmented book is the registration between the real and virtual world. To mitigate this issue, augmented books require visual tracking through a camera which recognizes the current page among numerous pages and calculates a 6 DOF pose in real time. However, visual tracking for an augmented book is quite difficult because a book generally includes tens or hundreds of pages and because tracking should be performed in real time (at more than 25fps).

To address these difficulties, most augmented books have thus far employed fiducial marker tracking methods [CLY07], [FYK*04], [BKP01], [JRP*05]. Fiducial markers are surrounded by a black rectangle or circle boundary for easy detection and include a bit pattern for ID representation. While the use of fiducial markers is a convenient and simple choice for an augmented book, these markers can lead to visual discomfort due to their distinctive shapes. Moreover, they are fragile to partial occlusion so that they are likely to distract users' immersion.

Recently, augmented books have employed markerless tracking methods that do not cause visual discomfort. With markerless tracking, a page pose is calculated from natural features extracted from the page without any fiducial markers [THKN07], [SPFL08]. Markerless tracking methods, however, involve fundamentally high computational costs compared to fiducial marker tracking. Moreover, they are associated with problems related to an increment in computational time and a drop in page recognition accuracy as the number of pages increases.

In addition, Yang et al. proposed a hybrid visual tracking method which merges the merits of fiducial marker track-

DOI: 10.2312/EGVE/JVRC09/013-020

ing and markerless tracking [YCS*08]. This work uses a tiny marker for page recognition and a randomized forest for page pose calculation.

For augmented books, this paper presents a markerless visual tracking method which recognizes the current page among numerous pages and calculates its 6 DOF pose in real-time. Given an image input by a camera, the proposed tracking method first recognizes a page and then performs wide-baseline keypoint matching to calculate its initial pose. For these purposes, a generic randomized forest (GRF) which extends the original randomized forest proposed in [LF06] is proposed. With the GRF, the tracking method can recognize and track pages without an increment in computational time and a drop in page recognition accuracy as the number of pages increases. Moreover, the tremendous memory consumption of the randomized forest referred to as the weakest point is reduced. Once a page is recognized and its initial pose is calculated, the tracking method performs the page tracking process without page recognition until the page is turned. The page tracking process selects a keyframe of the page adequate for tracking and employs a coarse-to-fine approach.

2. Related Work

There are various natural features for markerless tracking, such as keypoints, lines, eigen-images, and others. Keypoints tend to have less of a computation cost. Moreover, they are more robust with a range of viewpoint, illumination variations and partial occlusion compared to other features [LF05]. Therefore, this markerless tracking method for augmented books is based on keypoints and utilizes wide-baseline keypoint matching methods to recognize the current page and calculate its initial pose. In the following paragraphs, we look into wide-baseline keypoint matching methods, page recognition methods, and page tracking methods.

2.1. Wide-baseline Keypoint Matching

Most wide-baseline keypoint matching methods build an affine-invariant descriptor for each keypoint and match keypoints using the descriptors. To build affine-invariant descriptors, scale selection, rotation correction, and intensity normalization processes are required. The well-known methods are SIFT [Low04], GLOH [MS05], and SURF [BTV06]. However, these methods consume a considerable amount of time while building descriptors; hence, they are not suitable for augmented reality applications which require strong real time performance.

Recently, the keypoint matching method of Lepetit el al. [LF06] using a randomized forest (RF) is capable of real time performance and is robust to the viewpoint variations. Therefore, many researchers have worked with this method. They transform a local image patch of each keypoint into nearly possible appearances and train randomized trees in a RF with those transformed patches. The internal nodes of the trees test the intensity difference between two pixels of an image patch and leaf nodes store the posterior distribution of all keypoints. In [WKR07], Williams et al. proposed a modified RF which makes real-time training possible and applied this method to a real-time SLAM problem. They selected two points randomly to test at the internal nodes of the trees instead of choosing two points considering how much information can be gained.

As mentioned before, each node test of a RF is determined randomly. That implies the node tests are independent of training data. Accordingly, once the node tests of a RF for one training data set are set, it is possible to reuse those for another training data set. This property is referred to as reusability, and a generic randomized forest (GRF) is proposed that maximizes this type of reusability to the point that the GRF performs simultaneous page recognition and wide-based keypoint matching.

Although RFs represent a state-of-the-art real-time keypoint matching scheme, a RF for numerous pages requires a tremendous amount of memory - approximately 3.2GB for 100 pages. This much memory is unavailable for normal desktop computers and is therefore an obstacle to commercialization. Hence, this study proposes a method reducing memory consumption via the use of the property of FAST keypoints [RD06b].

2.2. Page Recognition

To our best knowledge, there have been two reports describing a page recognition method without fiducial markers. In [THKN07], Taketa et al. compared the input image to all model pages using an active search technique to recognize the current page and then selected the page with the highest similarity score as a recognition result. In [SPFL08], Scherrer et al. built a RF for each page during a training stage. They recognized a page by comparing keypoints extracted from the input image with those extracted from model pages using RFs.

As both systems only tracked a book with 7 to 10 pages and because precise experimental results were not presented, it is not an exact representation of this problem. However, it is possible to predict that the computation would linearly increase when tracking over a hundred of pages. Moreover, the page recognition accuracy of this scheme would fall.

2.3. Page Tracking

The various tracking methods can be classified into two groups according to whether or not prior information is given. These two groups are model-based tracking and SLAM-based tracking. In the model-based approach ([LF05], [RD06a]), tracking is mainly separated from the training stage. The training stage involves predefining objects which will be tracked during the tracking stage.

However, the SLAM-based approach ([DRMS07], [KM07]), it is capable of tracking in unknown environments because it performs both tracking and map building simultaneously, which is expensive in computation. For an augmented book, the model-based approach is more feasible because a training stage is necessary to recognize a page. Hence, it becomes possible to determine which objects are to be augmented on their proper pages. However, to reduce the burden of the training stage, an idea is borrowed from the SLAM-based approach. First, a relatively small number of keypoints(200) is extracted during the training stage. Second, additional keypoints (at most 1000 in total per page) are extracted from the initially selected keyframe during the tracking stage.

This indicates that both online and offline information is used for the page tracking process. This creates tracks with a considerable amount of information. This is described in Section 3.2 in detail.

3. Proposed Method

The proposed tracking method requires a training process prior to performing in real-time. The training process obtains one training image per page from a user and trains the generic randomized trees using the training images. Details of the training process are explained in Section 3.1.

In real-time, keypoints are first extracted from the input image because they are commonly used in the page recognition process and the page tracking process as shown in Figure 1. If there is no valid page ID, the page recognition process is executed; otherwise, the page tracking process is executed. An invalid page ID arises in case a page is not recognized or if page tracking fails at the prior frame. Once a page is recognized and its initial 6 DOF pose is calculated during the page recognition process, a valid page ID is created and the results including the page ID, rotation matrix (R), and translation vector (t) are conveyed to the page tracking process. The page tracking process is performed repeatedly until page tracking fails due to rapid page movement or a page turn. Details of the page recognition process are explained in Section 3.1 while those of the page tracking process are discussed in Section 3.2.

3.1. Page Recognition Process

The page recognition process is intended to recognize a page and calculate its initial 6DOF pose. For these purposes, the generic randomized forest plays a key role in both page recognition and keypoint matching. To calculate the initial pose of the recognized page, it is necessary to remove outliers among matching pairs and estimate the page pose relative to the camera. The PROSAC method [CM05] is used for the outlier removal and the method of Schweihofer et al. [SP06] is sused for pose estimations.

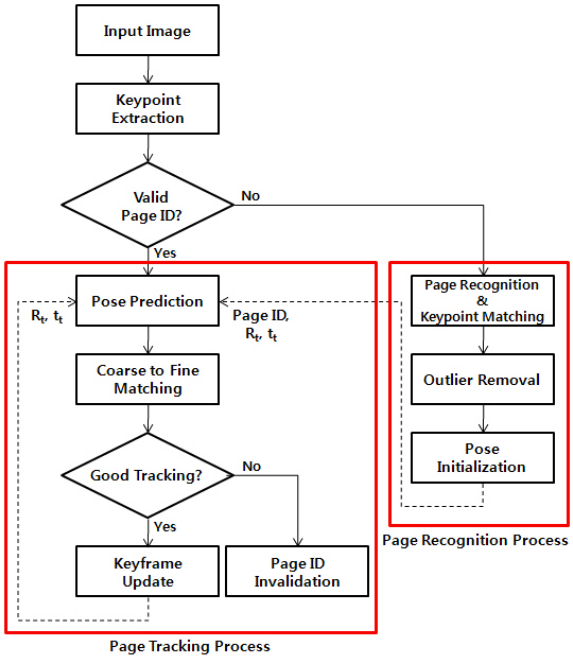

Figure 1: *Overview of markerless visual tracking*

3.1.1. Keypoint Extraction

The FAST detector [RD06b] was chosen for use because it is known to be very efficient for keypoint extraction. Keypoints are extracted from three octaves of an image to handle scale variations. To confirm whether each pixel p is a keypoint, this method considers 16 circular pixels at a distance of 3 from pixel p. If p is determined to be a keypoint and if p is brighter than its neighbors, p is referred to as a positive keypoint. If p is darker, it is a negative keypoint. In keypoint matching, if the keypoint p of the current image is positive, p does not need to be compared to negative keypoints of model images and vice versa. This property reduces the number of keypoint comparisons and the memory consumption of a randomized forest, which is considered as the weakest point. Additional details are given in section 3.1.2.

3.1.2. Generic Randomized Forest

The generic randomized forest (GRF) is utilized as it maximizes the reusability of a randomized forest (RF). Hence, it can perform page recognition and wide-baseline keypoint matching simultaneously. As mentioned in Section 2.1, each node test of a RF is determined in a random manner. This implies that the node tests are independent of the training data. Therefore, once node tests of a RF for keypoint matching within a certain page are built, it is possible to reuse those for another page. Thus it is possible to share the common node tests of one RF instead of building a RF for each page. Furthermore, if page recognition is designed well enough to be

performed using the common RF, the RF becomes generic; it can perform page recognition and wide-baseline keypoint matching. The RF is referred to as the generic randomized forest.

Training the GRF

The structure and the training method of a GRF are basically equivalent to the original RF apart from the probabilities stored in the leaf nodes, as shown in Figure 2. A GRF consists of NT randomized trees $T_1, T_2, \cdots, T_{NT}$, and all node tests of the GRF are built in a random manner. After the GRF is trained, every leaf node stores one probability distribution for page recognition and Nc probability distributions for keypoint matching for Nc pages.

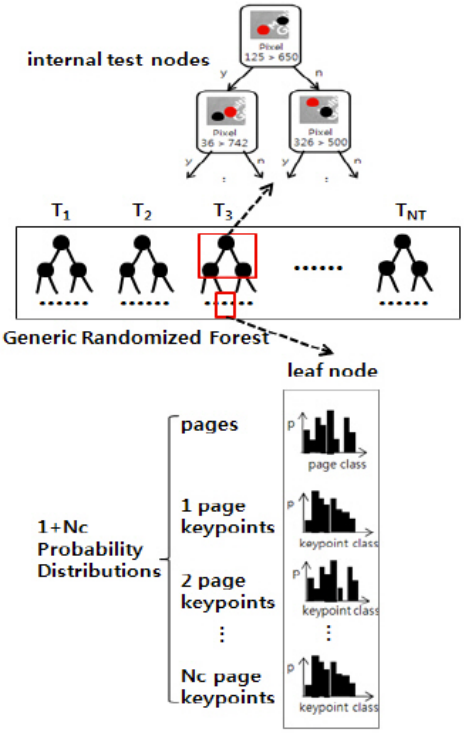

Figure 2: *Generic Randomized Forest*

To train the GRF for page recognition, new views of each page are first synthesized from the corresponding training image using randomly selected affine transformations. Keypoints are extracted from each new view by the FAST detector and are formed into the training data set for each page. In the prepared training data sets for all pages, every keypoint in the training data sets passes through all NT trees. If a keypoint from i-th page reaches the l-th leaf node $\xi_{t,l}$ of the t-th tree T_t, the frequency of page class i in $\xi_{t,l}$ increases. Finally, each leaf node stores the total visiting number of keypoints and the frequency of page classes. If the total visiting number in leaf node $\xi_{t,l}$ is $N_{t,l}$ and the frequency belonging to page class i is $N_{t,l,i}$, the posterior of page class i is calculated using (1).

$$P(C = i \mid \xi_{t,l}) = \frac{N_{t,l,i}}{N_{t,l}} \quad (1)$$

While training the GRF for keypoint matching within a page, new views of each keypoint extracted from the page are synthesized using affine transformations instead of transforming the entire image as in training for page recognition. These new views are formed into the training data set for each keypoint. In the prepared training data sets for all keypoints within the page, every keypoint in the training data sets passes through all NT trees. In the i-th page, if the total visiting number in leaf node $\xi_{t,l}$ is $N_{t,l}$ and the frequency belonging to keypoint class k is $N_{t,l,k}$, the posterior of keypoint class k is calculated using (2).

$$P(K = k \mid i, \xi_{t,l}) = \frac{N_{t,l,k}}{N_{t,l}} \quad (2)$$

This training process for keypoint matching is performed repeatedly with the other pages.

As a result of training, every leaf node of the GRF stores one probability distribution for page recognition and Nc probability distributions for keypoint matching for Nc pages as shown in Figure 2.

Page recognition using the GRF

Given an image taken by a camera in real-time, N keypoints are first extracted from the image by the FAST detector and then pass through all NT trees. One keypoint m_j reaches NT leaf nodes and giving NT probability distributions. In addition, the final probability distribution with respect to keypoint m_j can be acquired by considering their average. Finally, the page recognition result is obtained by considering the average of the final probability distributions with respect to all keypoints, as in (3).

$$\begin{aligned} Page\ \hat{i} &= argmax_i\, P(C = i \mid T_1, \cdots, T_{NT}, m_1, \cdots, m_N) \\ &= argmax_i\, \frac{1}{N} \sum_{j=1}^{N} \frac{1}{NT} \sum_{t=1}^{NT} P(C = i \mid leaf(T_t, m_j)) \end{aligned} \quad (3)$$

, where $leaf(T_t, m_j)$ is the leaf node which m_j reaches in T_t.

Wide-baseline keypoint matching using the GRF

After a page is recognized, keypoint matching is performed. However, in keypoint matching, the keypoints do not need to pass through all NT tress in the GRF once again because the structure of the GRF is shared in page recognition and wide-baseline keypoint matching. This point makes the proposed method very fast despite the fact that it performs both tasks. Thus, the GRF is appropriate for augmented reality applications which require recognition and tracking of a current object among numerous target objects. If the i-th page is recognized, 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 (4).

$$\begin{aligned} Keypoint\ \hat{k} &= argmax_k\ P(K=k \mid T_1,\cdots,T_{NT},m_j) \\ &= argmax_k\ \frac{1}{NT}\sum_{t=1}^{NT} P(K=k \mid leaf(T_t,m_j)) \end{aligned} \quad (4)$$

Reducing the memory consumption

The original RFs for numerous pages require a tremendous amount of memory to store posteriors. As an experiment result, it was determined that the most efficient RF had the number of trees as $NT = 40$, a depth of $d = 10$, and the number of keypoints $Nf = 200$ which requires 32MB. If a book consists of 100 pages, approximately 3.2GB is required. Thus, it is crucial to reduce the memory consumption of the RF before commercialization can be considered.

In [WRM*08], Wagner et al. reported that memory consumption can be reduced to 25% of its original burden without a performance loss by storing posteriors as 1- byte integer values instead of 4-byte floating values.

In addition, it is possible to reduce memory consumption at most by half using the positive/negative property of FAST keypoints. As mentioned in Section 3.1.1, if a keypoint of the input image is positive, it does not need to be compared to the negative keypoints of model images and neither do any negative keypoints. Thus, it is possible to separate the trees for positive keypoint matching from the trees for the negative instances in a RF according to the ratio of the number of positive keypoints (Nf_{pos}) to that of negative keypoints (Nf_{neg}). The trees for positive keypoints store the posterior distribution of only Nf_{pos} positive keypoints in its leaf node instead of storing the posterior distribution of all of the keypoints. This is also true for the trees of negative keypoints. Thus, the memory consumption reduces according to the ratio of Nf_{pos} to Nf_{neg} and at most by half in case Nf_{pos} is equal to Nf_{neg}. Furthermore, if both the method of Wagner et al. and the proposed method are applied to the RF, memory consumption can be reduced to 1/8 of the original burden.

3.2. Page Recognition Process

When a page is recognized or recovered from a tracking failure, the page is tracked by estimating the camera pose according to the motion model, projecting and matching the map points in two stages (coarse and fine), and finally, refining the camera pose from the matching pairs. This is described in 3.2.1 in detail.

When a new page is recognized by the page recognition process, the training image of the page is initially used as a keyframe to track the page. However, this is quite dangerous in terms of tracking stability because the training image may have been taken with a different camera in a different environment. Thus, the current frame is updated as a keyframe just in case it describes the page better than the existing keyframe according to the score calculated in section 3.2.2 after every instance of tracking success. In addition, more keypoints are extracted from the initial detected keyframe and they are added them to the world map for better tracking with abundant map points.

3.2.1. Tracking a page

This section describes a keypoint-based tracking process for a recognized page with the assumption that a 3D world map has already been constructed. At every frame, the following procedure is performed.

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 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. The motion model is updated.

Camera Motion

Camera motion M can be parameterized with a six-vector μ, a translation for the first three and rotation for latter elements, using an exponential map [Var74]. 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 via (5) [KM07].

$$\hat{P} = MP = exp(\mu)P \quad (5)$$

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

Patch search

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

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

, where x is a 2D point in an image coordinate, and K is the intrinsic matrix of the camera. Affine warping is performed to take account of viewpoint changes between the 8×8 image patch generated from the keyframe of the world map and the current camera position, as described in [KM07]. The determinant of the affine warping matrix is used to determine the pyramid level at which the patch can 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.

Coarse-to-fine matching

To make the page tracking process more robust to rapid camera motions, the patch search and pose update are done twice. First, a coarse search is done with only 60 map points from highest levels of the image pyramid of the current frame. A 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 [Hub81] of the reprojection error iteratively. With the refined pose, a fine search is done with up to 500 map points. At this point, the patch search is performed with a smaller search region. The final camera pose is eventually calculated and the camera motion is updated from the difference between the initial and final camera pose of the frame.

Tracking evaluation and failure recovery

Tracking is likely to fail in the occurrence of a motion blur, occlusion, or an incorrect position estimate. Thus, if a fraction of the result of keypoint matching falls below a certain threshold, it is considered as a tracking failure, causing the valid page ID bit to be set to false. Thus, the page recognition process will be performed to recover a camera pose at the next frame.

3.2.2. Keyframe update

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

1. An image is clear enough with no motion blur.
2. The area of the 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 direction are orthogonal.

The total score function of the t-th frame is the weighted sum of the three subscore functions of $Score_{ZMSSD}$, $Score_{area}$, and $Score_{ortho}$, as shown in (7).

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

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

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

$Score_{area}$ measures the portion of the pages are shown within the image at frame t, as shown in (9).

$$Score_{area}(I_t) \;=\; \frac{Area_t}{ImageSize}\,\frac{Area_t}{AreaOfPage_t} \tag{9}$$

, where $Area_t$ denotes the area of the page shown in the image at frame t in pixel scale and $AreaOfPage_t$ is the area of the page including the area beyond the image boundary after four boundary points of the page are projected into the image according to the camera pose. $Score_{ortho}$ measures how orthogonal the page is to the camera z vector, by comparing the z coordinate of the camera z vector($CamZ_{z,t}$) and the normal vector of the page ($PageNorm_{z,t}$) after projecting it into the camera coordinate according to the camera pose, as shown (10).

$$Score_{ortho}(I_t) \;=\; 1 \;-\; \left|CamZ_{z,t} \;+\; PageNorm_{z,t}\right| \tag{10}$$

Therefore, the higher $Score_{total}$ is, the more accurately the page tracking process tracks the pose of a page.

4. Experiment Result

For the experiment, a laptop with a 2.2GHz Core 2 Duo CPU with 2GB memory and an ATI Mobility Radeon HD 2400 graphic card were used. A Logitech Ultra camera was attached to the laptop. A 640x480 image was obtained from the camera and keypoints were extracted using the FAST detector in real-time.

4.1. Performance of the page recognition process

The experiments were performed using the GRF with the number of trees at $NT = 40$ and a depth of $d = 10$. First, the recognition performance of the GRF was evaluated as it affects the overall performance. For the experiment, the GRF with 20 pages and nine test images taken from different viewpoints were prepared for each page, and a total 180 images were taken for the evaluation. To determine how many keypoints per page suitably describe a page in training, the recognition rate was examined as the number of keypoints increased from 10 to 300. Figure 3 shows the experimental result.

160 keypoints recorded the highest recognition rate, 89.2% although there was only slight difference with more than 100 keypoints. In fact, the recognition rate was not as high as expected. However, if erroneous page recognition occurs, the proposed method ignores the page through outlier removal during the page recognition process. Thus, false positives generally do not occur.

4.2. Performance of the page tracking process

To show that a page is being tracked correctly, the world model of a page (red rectangle) and the recognized page ID (page number) are projected onto the image according to the camera pose and a virtual object assigned to each page is augmented on the page.

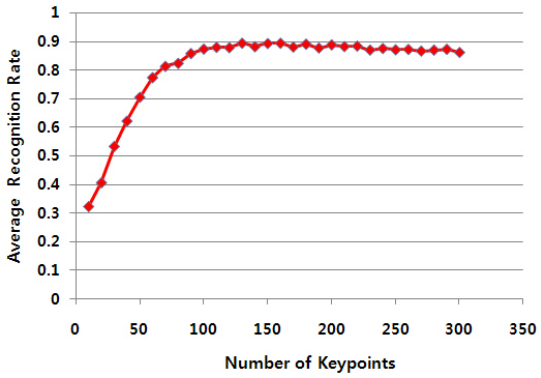

Figure 3: *Page recognition rate with respect to the number of keypoints*

Figure 4 shows that the page tracking process works well with dramatic viewpoint variations of the camera(first row), scale variations (second row) by moving the camera back and forth, illumination variations (third row), partial occlusions (fourth row), and complicated environments in which tracking is likely to be disturbed (fifth row).

Figure 5 shows the tracking time for a book of 11 pages. The average tracking time for 1270 frames was 8.61 ms. This is represented as a straight red line in the graph. Most timing spikes occur when the page recognition process is attempting to recognize a page, but the spike that takes place around frame 205 was due to a tracking failure.

5. Conclusion

For augmented books, this paper presents a markerless visual tracking method which recognizes the current page among numerous pages and estimates its 6 DOF pose in real-time. The results show that the average tracking time for 1270 frames is 8.61 ms. The GRF used to recognize a page ID takes at most 35ms during experiments, which implies that it guarantees real-time execution at almost 30 fps while initially showing 90% accuracy in recognition. Thus, although tracking fails, the tracking process recovers very quickly within several frames. However, the accuracy of recognition decreases if the environment of a page in an image is disordered. The page tracking process with the online and offline information and the keyframe update demonstrates good tracking results. However, because it uses a motion model when it predicts the pose at every frame, it tends

Figure 4: *Tracking in situations with dramatic viewpoint variation, scale variations, illumination variations, partial occlusions, and complicated environments*

to fail to track a page when the page shows irregularly dramatic movements. Thanks to the page recognition process, tracking then restarts within a few frames.

Therefore, additional experiments related to the accuracy of page recognition and page poses can be performed as a future work. This may improve the aforementioned.

6. Acknowledgements

This research is supported by the Ubiquitous Computing and Network(UCN) Project, Knowledge and Economy Frontier R&D Program of the Ministry of Knowledge Economy(MKE) in Korea and a result of subproject UCN 09C1-J2-11T. Authors are gratefully acknowledging the financial support by Agency for Defense Development and by UTRC(Unmanned technology Research Center), Korea Advanced Institute of Science and Technology.

References

[BKP01] BILLINGHURST M., KATO H., POUPYREV I.: The magicbook - moving seamlessly between reality and virtuality. *IEEE Computer and Graphics Application 21*, 3 (Jan. 2001), 6–8.

[BTV06] BAY H., TUYTELAARS T., VANGOOL L.:

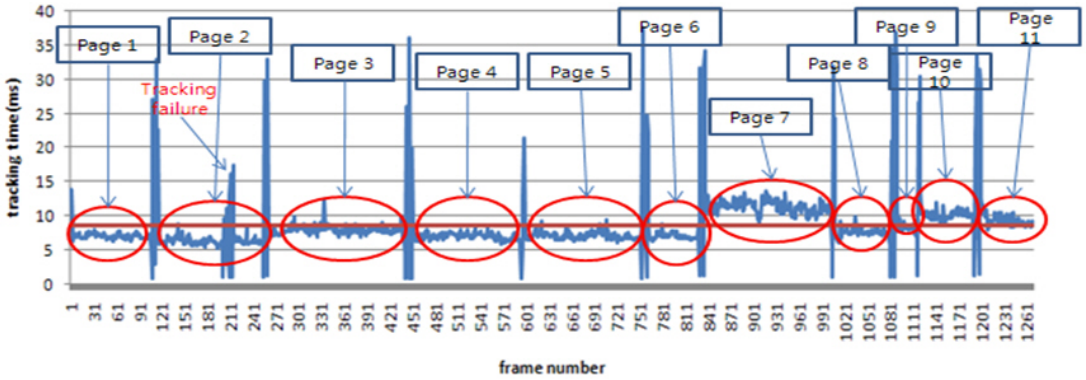

Figure 5: *Tracking time(left axis) for a book of 11 pages. The timing spikes occur when tracking is lost, thus the page recognition process is attempted*

SURF:Speeded Up Robust Features. In *Proc. 9th European Conference on Computer Vision* (2006), pp. 404–417.

[CLY07] CHO K., LEE J., YANG H. S.: A Realistic e-Learning System based on Mixed Reality. In *Proc. 13th International Conference on Virtual Systems and Multimedia* (2007), pp. 57–64.

[CM05] CHUM O., MATAS J.: Matching with PROSAC-Progressive Sample Consensus. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (2005), pp. 220–226.

[DRMS07] DAVISON A. J., REID I. D., MOLTON N. D., STASSE O.: MonoSLAM: Real-time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 6 (2007), 1052–1067.

[FYK*04] 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).

[Hub81] HUBER P.: *Robust Statistics*. Wiley, 1981.

[JRP*05] JUAN M. C., REY B., PEREZ D., TOMAS D., ALCA N. M.: The memory book. In *Proc. 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology* (2005), pp. 379–380.

[KM07] KLEIN G., MURRAY D.: Parallel tracking and mapping for small AR workspaces. In *6th IEEE and ACM International Symposium on Mixed and Augmented Reality* (2007), pp. 255–234.

[LF05] LEPETIT V., FUA P.: Monocular Model-Based 3D Tracking of Rigid Objects: A Survey. *Computer Graphics and Vision 1*, 1 (2005), 1–89.

[LF06] LEPETIT V., FUA P.: Keypoint Recognition using Randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*, 9 (2006), 1465–1479.

[Low04] LOWE D. G.: Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision 08*, 2 (2004), 91–110.

[MS05] MIKOLAHCZYK K., SCHMID C.: A Performance Evaluation of Local Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27*, 10 (2005), 1615–1630.

[RD06a] REITMAYR G., DRUMMOND T.: Going out: Robust Model-based Tracking for outdoor Augmented Reality. In *6th IEEE/ACM International Symposium on Mixed and Augmented Reality* (2006), pp. 109–118.

[RD06b] ROSTEN E., DRUMMOND T.: Machine learning for high-speed corner detection. In *Proc. 9th European Conference on Computer Vision* (2006), pp. 430–443.

[SP06] SCHWEIGHOFER G., PINZ A.: Robust Pose Estimation from a Planar Target. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*, 12 (2006), 2024–2030.

[SPFL08] SCHERRER C., PILET V., FUA P., LEPETIT V.: The haunted book. In *Proc. 7th IEEE/ACM International Symposium on Mixed and Augmented Reality* (2008).

[THKN07] TAKETA N., HAYASH K., KATO H., NISHIDA S.: Virtual pop-up book based on augmented reality. In *Proc. HCI* (2007), pp. 475–484.

[Var74] VARADARAJAN V.: *Lie Groups, Lie Algebras and Their Representation*. SpringerVerlag, 1974.

[WKR07] WILLIAMS B., KLEIN G., REID I.: Real-Time SLAM Relocalisation. In *Proc. 11th IEEE International Conference on Computer Vision* (2007), pp. 1–8.

[WRM*08] WAGNER D., REITMAYR G., MULLONI A., DRUMMOND T., SCHMALSTIEG D.: Pose Tracking from Natural Features on Mobile Phones. In *6th IEEE/ACM International Symposium on Mixed and Augmented Reality* (2008), pp. 125–134.

[YCS*08] YANG H. S., CHO K., SOH J., JUNG J., LEE J.: Hybrid Visual Tracking for Augmented Books. In *Proc. International Conference on Entertainment Computing* (2008), pp. 161–166.