

Virtualized reality model-based benchmarking of AR/MR camera tracking methods in TrakMark

Koji Makita*
AIST, Japan

Takashi Okuma†
AIST, Japan

Tomoya Ishikawa‡
AIST, Japan

Laurence Nigay§
Université J. Fourier, France

Takeshi Kurata¶
AIST, Japan

ABSTRACT

We describe a virtualized reality model-based benchmarking of camera tracking methods for augmented reality and mixed reality. Benchmark datasets created with virtualized reality models do not include any measurement errors, and can be effectively used for a benchmark of tracking methods [5][6]. But there are many issues to be discussed for creators and users of the datasets. For the creators, efficient creation and sharing of the datasets are priority issues. On the other hand, supports for benchmarking processes and for creating original datasets are desired by the users. To tackle these issues, we propose a prototype framework for a virtualized reality model-based benchmarking of camera tracking methods. In this paper, first, we describe an outline of the framework, and show actual components in the framework. Moreover, we show experimental results of benchmarking our tracking method with the framework.

Index Terms: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented and virtual realities; H.5.2 [Information Interfaces and Presentation]: User Interfaces—Benchmarking;

1 INTRODUCTION

In a research field of image processing, ground truth data play a great role to evaluate and improve visual tracking methods. For example, “The Yosemite Sequences” [2] has been widely used and studied for evaluating estimation results of optical flow [3]. Moreover, Lieberknecht et al have proposed a dataset and evaluation methodology for template-based tracking algorithms [1]. Endres et al have proposed an evaluation for RGB-D SLAM systems with a dataset that contains camera images with time-synchronized ground truth poses obtained from a high-accuracy motion-capture system [4].

In a working group “TrakMark” [8], we held various activities about evaluations of tracking methods for augmented reality and mixed reality [9][10]. One of our main activity is to create and share benchmark datasets for camera tracking methods. To create the datasets, ground truth of camera parameters precisely correlating to images are needed. But making ground truth data is costly in real environment. In order to make ground truth data, for example, in [1], we have to use a camera with accurately controlled devices like a robot arm to get movement data. On the other hand, in [4], we have to set up motion-capture systems in environments.

In this paper, we describe virtualized reality model-based benchmark of camera tracking methods for augmented reality and mixed reality. By applying the models, we can create ground truth data without any measurement errors. However, there are many issues

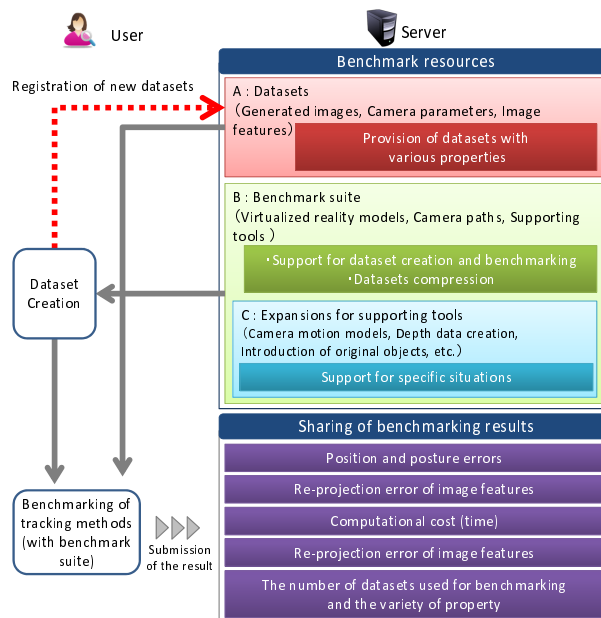


Figure 1: An outline of our proposed framework

to be discussed for creators and users of the datasets. For creators, efficient creation and sharing of the datasets are priority issues. On the other hand, a tool for supporting benchmarking processes and for creating original datasets is desired by users. For the creators and the users, we propose a prototype framework for a virtualized reality model-based benchmarking of camera tracking methods. In this paper, first, we describe an outline of the framework. Second, we show actual components in the framework. Finally, we show experimental results of benchmarking our tracking method with the framework.

2 FRAMEWORK OF BENCHMARKING

Figure 1 shows an outline of our proposed framework. The framework is targeted at two types of benchmarkings. The first one is easy and fair benchmarking of camera tracking methods for comparison purposes (official evaluation). The second one is each user’s own benchmarking (local evaluation). In the following, we describe how the user utilizes the framework for comparison purposes and for own benchmarking.

2.1 Benchmarking for comparison purposes

In benchmarking process for comparison purposes, first, the user downloads datasets (in Figure 1, “A : Datasets”), and creates estimated camera parameters with his or her tracking methods. The datasets includes image sequences, ground truth of camera parameters, and image features. Next, the user sends benchmarking results to the server. After that, benchmarking results are shared in the server. Finally, the user can compare his or her results with other results.

Benchmarking results on same datasets and same indexes can be easily and fairly compared. Therefore, standardizations of bench-

*e-mail: k.makita@aist.go.jp

†e-mail: tomoya-ishikawa@aist.go.jp

‡e-mail: takashi-okuma@aist.go.jp

§e-mail: laurence.nigay@imag.fr

¶e-mail: t.kurata@aist.go.jp

mark indexes and formats of the datasets are helpful. Now we are planning to standardize indexes for benchmarking camera tracking method for AR/MR with a standardization of dataset-format in “ISO / JTC 1 / SC 24 / WG 9 (Augmented reality continuum concepts and reference model)” [13]. In the server, we treat five types of benchmark indexes (B.I.) below.

- B.I.1: Position and posture errors
- B.I.2: Re-projection error of image features
- B.I.3: Computational cost (time)
- B.I.4: Projection error of virtual objects
- B.I.5: The number of datasets used for benchmarking and the variety of property

B.I.1, B.I.2, and B.I.3 are normal indexes in evaluating camera tracking methods. Usually, for calculating re-projection error of image features, a target object for projection is needed. For example, in [1], each sequence is focused on one picture, and four points around the picture are used. But in our datasets, camera can move widely in virtualized reality models, so that a few points for projection are not enough. Therefore, tracking data of interest points are included in our datasets [7]. Tracking data are composed of 3D-2D correspondences of interest points. 2D positions of interest points on generated images are calculated by using camera parameters and 3D positions of interest points.

B.I.4 is introduced as an index especially for AR/MR tracking methods. In calculating a projection error of virtual object, first, 3D position of the virtual object is set in front of the ground truth of camera position. Next, 2D positions of the virtual object on an image plane with ground truth of camera parameters and on an image plane with estimated camera parameters are calculated. Finally, a euclidean distance between the two positions is calculated as a projection error of the virtual object.

B.I.5 is introduced for preventing fine tuning of the tracking method only for some specific datasets. For applying this index, what is important is to prepare datasets with various kind of properties. Until now, we have uploaded multiple datasets with various kind of virtualized reality models shown in Figure 2. In future, we are planing to expand our datasets with property information. In TrakMark, as shown in Figure 3, a part of datasets with real images have already been classified with properties: “Significant moving occluders”, “Fast camera movement”, “Auto luminance control”, “Auto focus control”, and “Reference data”.

Supporting tools included in “Benchmark suite” (in Figure 1, B) have a function for users to support benchmarking tracking methods with standardized indexes. The construction of this function is our future work after fixing standardized indexes. Relating to the study of the tools, there are still issues about standardizations of benchmark process. First, a process to make benchmark results is to be discussed. Major issues of this process are how to get benchmark indexes and how to choose datasets with considering the number and property of datasets. Second, a process to share benchmark results in the framework is to be discussed.

2.2 User’s own benchmarking

Second motivation of the framework is to support each user’s own benchmarking (local evaluation). For achieve this goal, “Benchmark suite” and “Expansions for supporting tools” (in Figure 1, B and C) are prepared. A benefit of the user’s own benchmarking is that the user can create and use original datasets with considering the user’s own situations. The benchmark suite also contribute to compressing the datasets, because the user can generate preexistent datasets with the benchmark suite. Data size of the benchmark suite is very small compared to the datasets.

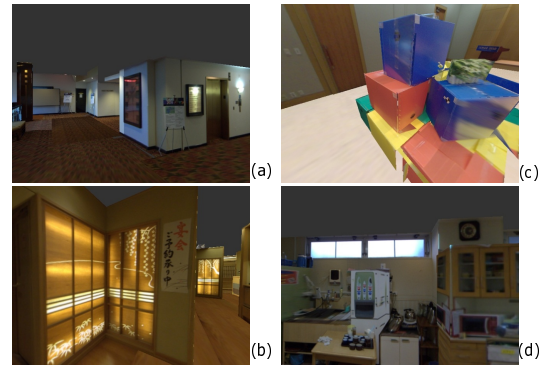


Figure 2: Examples of datasets created with virtualized reality models in TrakMark. (a): Conference Venue Package 01. (b): Conference Venue Package 02. (c): Japanese Restaurant Package 01. (d): Nursing Home Package 01.

Property of Image Sequences.		00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17
Properties ?	Seq. no.																		
Significant moving occluders											✓	✓	✓						
Fast camera movement															✓				
Auto luminance control																✓	✓	✓	✓
Auto focus control																			✓
Reference data		✓									✓			✓	✓				

Sampled Frames.

Figure 3: Datasets classified with properties. An upper table shows properties of image sequences in “NAIST Campus Package 02”. Lower images are sampled frames from sequences 9, 10, and 11 that include significant moving occluders.

What we are mostly concerned with the supporting tools is about how to efficiently create camera paths. Therefore, in actual benchmark suite, a tool with a function for efficiently creating camera paths is included. The tool is constructed based on our previous method [7]. Figure 4 shows an outline of creating datasets with the tool. By using the tool, the user can freely choose a virtualized reality model and set camera paths. The user sets control points on the ground plane, and camera parameters are automatically generated with linear interpolation. In the same time, the user can generate data of interest points. This data is used for calculating re-projection errors of image features.

In the framework, supporting tools can be used with expansions below.

- Application of camera motion models for camera paths
- Depth data creation
- Introduction of original objects

Generally, when we create datasets, there are many kinds of supposed scenarios. For example, human navigation, tabletop and desktop AR/MR, and so on. And the type of camera motions we want to represent depends on the scenario. In human navigation fields, for example, the camera is often supposed as hand-held type or head-mounted type that include various kind of shakes with human translation. But it takes many time to independently set all extrinsic parameters with considering shake of the hand or the head. Therefore, a function to apply camera motion models to the created camera path is included in the expansions. In actual components, walking motion model based on [12] is introduced for simulating a motion of head-mounted camera. The user of the tool can set six parameters shown in Figure 5 to control the motion. In future,

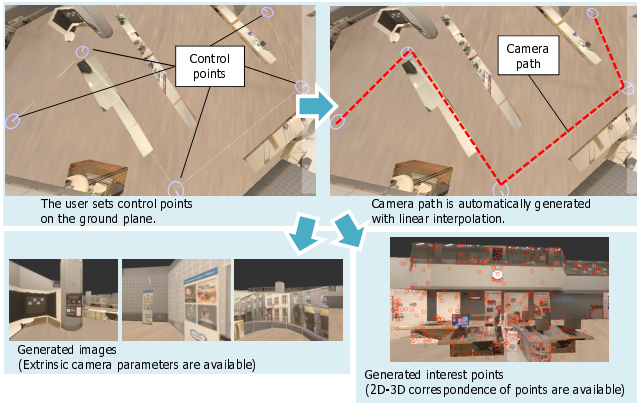


Figure 4: An outline of creating data sets

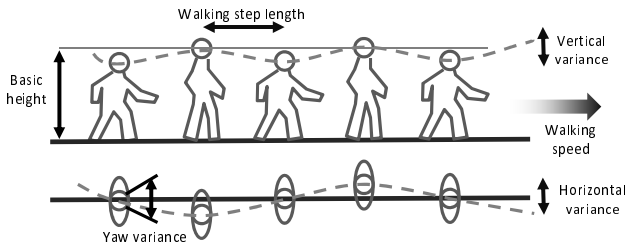


Figure 5: Parameters of walking motion model

we plan to add other kind of motion models for supporting various scenarios.

Depth data creation is for applications or researches using RGB-D sensors. In actual expansions, a function of creating depth data has already been included. Figure 6 shows examples of a generated image and corresponding depth image.

An introduction of user's original objects is an important function. For example, additional visual markers are effective for marker-based tracking methods. Until now, we can add visual markers to the model with our modeling tool [11]. Figure 7 shows examples of visual markers added in the model (the markers are created by Ritsumeikan university). In future, we plan to integrate this function into the supporting tools. Moreover, we also plan to introduce moving objects for tracking methods that considering moving occluders.

3 BENCHMARKING OF A TRACKING METHOD

To study indexes of the benchmarking, we conducted experiments of benchmarking a tracking method with our proposed framework. We created a dataset with human walking motion model of head-mounted camera as shown in Figure 8. In Figure 8, (A) shows an overview of the model and control points the user set on the ground plane of the model, (B) shows camera path created by the tool. (C), (D), and (E) are examples of generated images. The dataset includes 349 of generated images at 640×480 pixels without distortion and warping. The dataset includes a principal point (C_x, C_y) and focal lengths expressed in pixel-related units (F_x, F_y) as intrinsic parameters of generated images. In the dataset, intrinsic parameters stay constant, $(C_x, C_y) = (320, 240)$, and $(F_x, F_y) = (240, 240)$. It took about 10 minutes for setting control points and creating the dataset.

For this experiment, as a target of benchmarking, we developed a camera tracking method based on key-frame matching and interest point tracking. Next, we created estimated camera parameters with the dataset. Finally, we applied an index of "B.I.4: Projection error of virtual objects". Figure 9 shows an outline of calculating

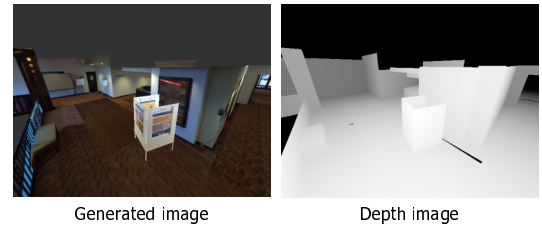


Figure 6: Examples of a generated image and corresponding depth image

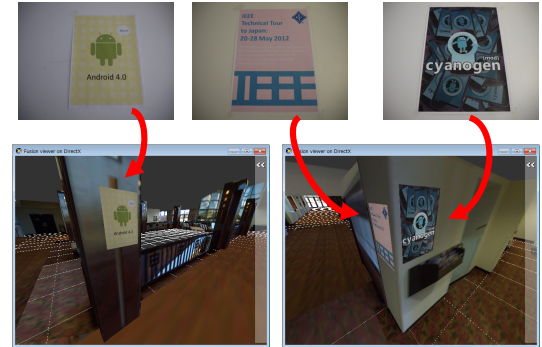


Figure 7: Visual markers added in the model (the markers are created by Ritsumeikan university).

the projection errors, and Figure 10 shows 3D coordinates of virtual objects. In the experiment, for studying variances of projection errors on the image plane, we prepared nine points of virtual objects on a virtual plane at a place of $z = a$ (a is a constant value).

Figure 11 and Figure 12 show experimental results in case $a = 1000 [mm]$ and $a = 3000 [mm]$, respectively. On the whole, the errors are larger when $a = 1000 [mm]$ (please notice that scales of projection errors in Figure 11 and Figure 12 are different). On the other hand, we observe that there are variances of projection errors among nine points. Especially, errors in the left side (point A, D, and G) are large. These results indicate that in case positions of virtual objects are precisely decided, camera tracking methods can be effectively evaluated with the projection errors. For example, researchers or developers of camera tracking methods can tune their methods with data of virtual objects. The users can compare abilities of camera tracking methods with their data of virtual objects.

Because we used only nine points on the plane as a first step of calculating projection errors, we could visualize all the errors with graphs shown in Figure 11 and Figure 12. But we have a possibility that the number of the points in the experiments was not enough to figure out true performances of tracking methods. In future, we plan to introduce more points and virtual planes, and statistically study relations between distribution of virtual objects and interest points detected by camera tracking methods.

4 CONCLUSION AND FUTURE WORK

As one of activities in TrakMark, we proposed virtualized reality model-based benchmarking of camera tracking methods for augmented reality and mixed reality. Because we create datasets with models, we can efficiently support various situations of tracking methods. In future, we plan to elaborating components of the framework with standardizations for indexes of the benchmarking processes and formats of the datasets.

ACKNOWLEDGEMENTS

This work was supported by Strategic Japanese-French Cooperative Program on Information and Communications Technology Including Computer Sciences (ANR and JST).

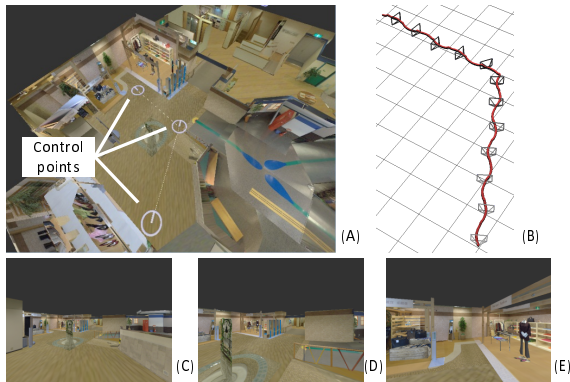


Figure 8: A dataset with human walking motions

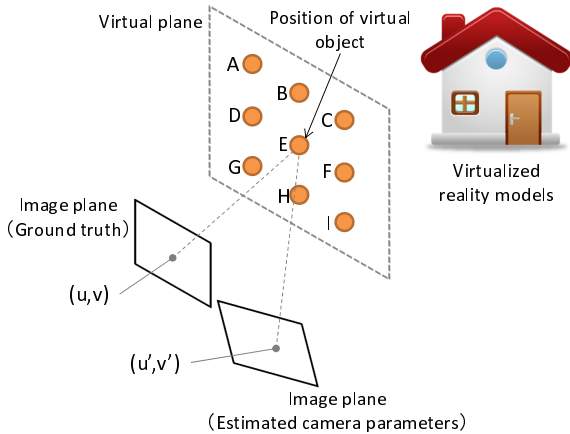


Figure 9: Projection error of virtual objects. (In every frame, relative positions between an image plane of ground truth and a virtual plane is fixed.)

REFERENCES

- [1] S. Lieberknecht, et al: "A Dataset and Evaluation Methodology for Template-based Tracking Algorithms", In ISMAR2009, pp.145-151, 2009.
- [2] D. J. Heeger: "Model for the extraction of image flow", Journal of the Optical Society of America A, vol.4, No.8, pp.1455-1471, 1987.
- [3] I. Austvoll: "A Study of the Yosemite Sequence Used as a Test Sequence for Estimation of Optical Flow", In SCIA2005, pp.659-668, 2005.
- [4] F. Endres et al: "An Evaluation of the RGB-D SLAM System", In Proc. of ICRA2012, 2012.
- [5] K. Hayashi, et al: "A Study of Camera Tracking Evaluation on TrakMark Data-Set", In Proc. of TrakMark2011, 2011.
- [6] A. Petit, et al: "Evaluation of Model based Tracking with TrakMark Dataset", In Proc. of TrakMark2011, 2011.
- [7] K. Makita, et al: "A prototype tool in TrakMark to generate benchmark data sets for visual tracking methods using virtualized reality models", In Proc. of TrakMark2011, 2011.
- [8] TrakMark WEB
<http://trakmark.net/>
- [9] H. Tamura, et al: "Proposal of International Voluntary Activities on Establishing Benchmark Test Schemes for AR/MR Geometric Registration and Tracking Methods", In Proc. of ISMAR2009, pp.233-236, 2009
- [10] F. Shibata, et al: "An Intermediate Report of TrakMark WG ", In S&T Proc. of ISMAR2010, pp.298-302, 2010.
- [11] T. Ishikawa, et al: "In-Situ 3D Indoor Modeler with a Camera and Self-Contained Sensors", In Proc. of HCI2009, pp.454-464, 2009.
- [12] E. Hirasaki, et al: "Effects of walking velocity on vertical head and body movements during locomotion", In Exp. Brain Res., vol.127, pp.117-130, 1999.
- [13] ISO (the International Organization for Standardization)/JTC1/SC24:
http://www.iso.org/iso/iso_technical_committee?commid=45252

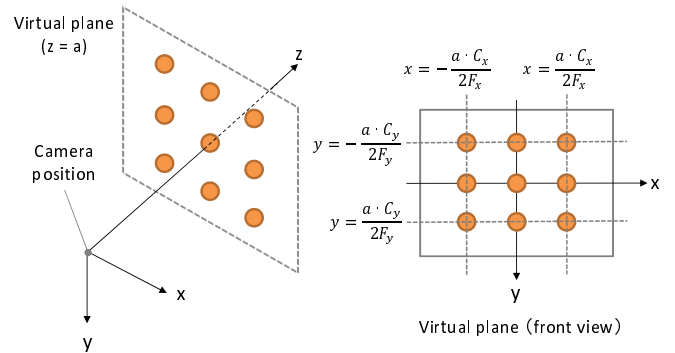


Figure 10: Positions of virtual objects.

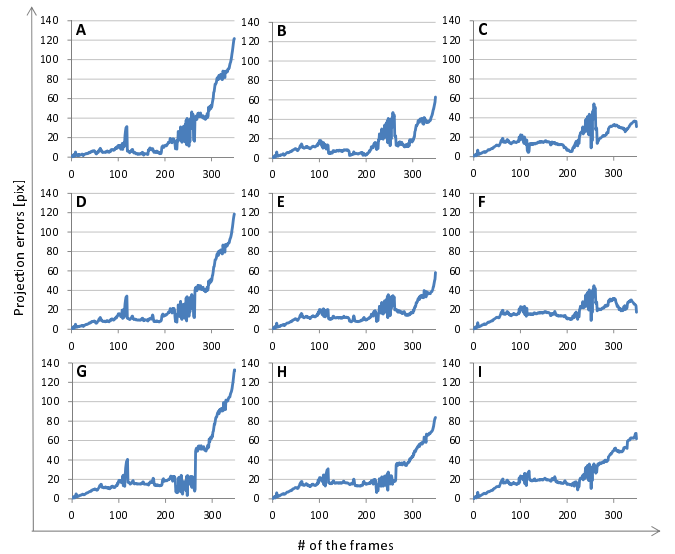


Figure 11: Projection error of virtual objects ($z = 1000 [mm]$)

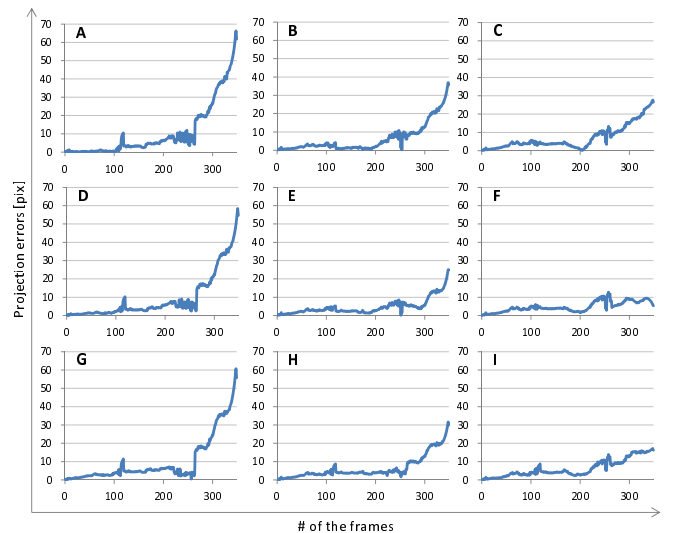


Figure 12: Projection error of virtual objects ($z = 3000 [mm]$)