A benchmark suite with virtualized reality models for supporting tracking evaluation and data set generation

Koji Makita
AIST, Japan
k.makita@aist.go.jp

Takashi Okuma
AIST, Japan
takashi-okuma@aist.go.jp

Tomoya Ishikawa
AIST, Japan
tomoya-ishikawa@aist.go.jp

Laurence Nigay
Université J. Fourier, France
laurence.nigay@imag.fr

Takeshi Kurata
AIST, Japan
t.kurata@aist.go.jp

Abstract

We describe a benchmark suite with virtualized reality models for augmented reality and mixed reality. Benchmark datasets created with virtualized reality models do not include any measurement errors. On the other hand, supports for benchmarking processes and for creating datasets are desired by creators and users. The benchmark suite is for supporting tracking evaluation and data set generation. In this paper, we describe a design of the benchmark suite, and show experimental results of benchmarking our tracking method and creating datasets with the benchmark suite.

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, there are “The Yosemite Sequences” [2] for evaluating estimation results of optical flow [3], a dataset for template-based tracking [1], and a dataset for RGB-D SLAM systems [4]. In a working group “TrakMark” [8][9], we are studying a creation of benchmark datasets composed of virtualized reality models [7]. By applying the models, we can create ground truth data without any measurement errors, and can be effectively used for a benchmark of tracking methods [5][6]. On the other hand, a tool for supporting benchmarking processes and for creating datasets is desired by creators and users.

In this paper, we propose a benchmark suite for a virtualized reality model-based benchmarking of AR / MR camera tracking methods. In this paper, first, we describe an outline of our supposed framework for the benchmarking. Second, we show actual components of the benchmark suite. Finally, we show experimental results of benchmarking our tracking method and creating datasets with the benchmark suite.

2. Framework of benchmarking

Figure 1 shows an outline of our supposed 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”). The datasets includes image sequences, ground truth of camera parameters, and image features. Next, the user estimates camera parameters with his or her tracking methods, and 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 indices can be easily and fairly compared. Therefore, standardizations of benchmark indices and formats of the datasets are helpful. We plan to standardize indices 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)” [12]. In the server, we treat five types of benchmark indices (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

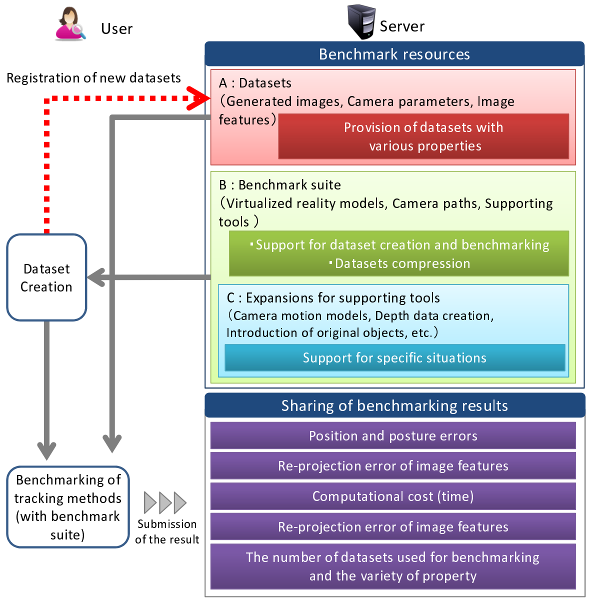

User
Server
Registration of new datasets
Benchmark resources
A : Datasets
(Generated images, Camera parameters, Image features)
Provision of datasets with various properties
B : Benchmark suite
(Virtualized reality models, Camera paths, Supporting tools)
・Support for dataset creation and benchmarking
・Datasets compression
C : Expansions for supporting tools
(Camera motion models, Depth data creation, Introduction of original objects, etc.)
Support for specific situations
Dataset Creation
Sharing of benchmarking results
Position and posture errors
Re-projection error of image features
Computational cost (time)
Re-projection error of image features
The number of datasets used for benchmarking and the variety of property
Benchmarking of tracking methods (with benchmark suite)
Submission of the result

Figure 1. An outline of our proposed framework

B.I.1, B.I.2, and B.I.3 are indices normally used for evaluating camera tracking methods. On the contrary, B.I.4 is introduced as an index especially for AR/MR tracking methods. In a calculation process of 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 an over-evaluation of the tracking method caused by fine tunings only for some specific datasets. For applying this index, it is important to prepare datasets with various kind of properties. Until now, we have uploaded four packages of datasets with six types of virtualized reality models. We plan to expand our datasets with property information. In TrakMark, 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 to benchmarking tracking methods with standardized indices. Current implementation of the function is tentative because the standardization of indices is still on-going. 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 indices 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.

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]. 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

A function to apply camera motion models to the created camera path is included for efficiently create natural camera path. In actual components, walking motion model based on [11] is introduced for simulating a motion of head-mounted camera. The user of the tool can set six parameters (basic height, walking step length, vertical variance, yaw variance, horizontal variance, and waking speed). Figure 2 shows examples of datasets created with the benchmark suite, (A) shows an overview of the model and control points the user set on the ground plane of the model, (B) and (C) show camera paths without and with the walking motion, (D), (E) and (F) are examples of generated images. 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.

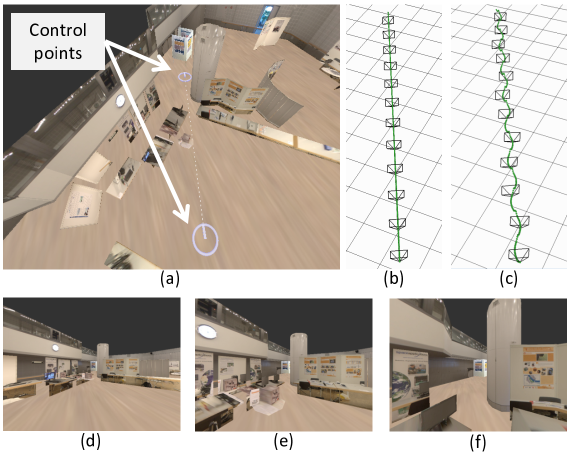


Figure 2. Examples of created datasets

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 [10]. We plan to integrate this function into the supporting tools, and also plan to introduce moving objects for tracking methods that considering moving occluders.

3. Benchmarking of a tracking method

We conducted experiments of benchmarking a tracking method with the benchmark suite. We created two datasets without and with human walking motion model of head-mounted camera as shown in Figure 2. Each dataset includes 356 of generated images at 640×480 pixels without distortion and warping, and also 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 both datasets, intrinsic parameters stay constant, $(C_x, C_y) = (320, 240)$, and $(F_x, F_y) = (240, 240)$. It took about five minutes for setting control points and creating both datasets. 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 indices of "B.I.1: Position and posture errors", and "B.I.4: Projection error of virtual objects".

In calculating position and posture errors, we used criteria shown in [6] with a 3×4 transformation matrix that is composed of 3×3 rotation matrix R and 3×1 translation matrix T. R_g and T_g show matrices of the ground truth, and R_e and T_e show matrices of the estimation. The position error E_T is calculated such that $E_T = ||T_g - T_e||$. The posture error E_P is calculated such that $E_P = \arccos((\mathrm{tr}(R_d) - 1)/2)$, where $R_d = R_g R_e^T$.

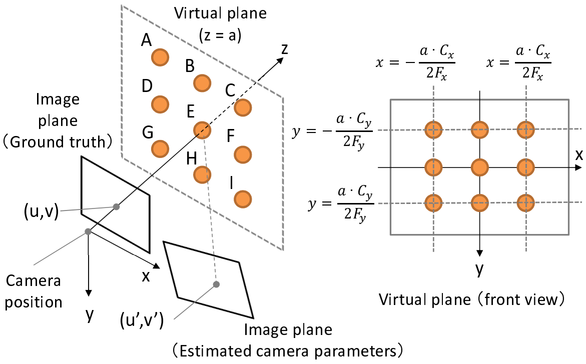

Virtual plane
(z = a)
A
B
C
D
E
F
G
H
I
z
Image plane
(Ground truth)
(u,v)
Camera position
x
y
(u',v')
Image plane
(Estimated camera parameters)
$x = -\frac{a \cdot C_x}{2F_x}$
$x = \frac{a \cdot C_x}{2F_x}$
$y = -\frac{a \cdot C_y}{2F_y}$
$y = \frac{a \cdot C_y}{2F_y}$
x
y
Virtual plane（front view）

Figure 3. Positions of virtual objects

In calculating projection errors, for investigating variances of projection errors, we prepared nine points of virtual objects on a virtual plane at a place of $z = a$ (a is variable to change the plane's position). Figure 3 shows 3D coordinates of virtual objects.

Figure 4 shows position and posture errors, and Figure 5 shows average projection errors. When the dataset without walking motion was applied, the tracking failed at 320th frame. On the contrary, when the dataset with walking motion was applied, errors drastically increased at about 250th frame because the application continued the tracking with large errors and didn't re-initialize the estimation. We think that these kinds of knowledge should be included in the benchmarking results, and can be used for improving camera tracking algorithms.

Average projection errors show a change similar to the position and posture errors in both without and with walking motions. On the other hand, as shown in Figure 6, we observe that there are significant variances of projection errors among nine points when $z = 0.5[m]$. Especially, errors in the right side (point B and C) with walking motion 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. 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 a benchmark suite for supporting tracking evaluation and data set generation. 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 indices of the benchmarking processes and formats of the datasets.

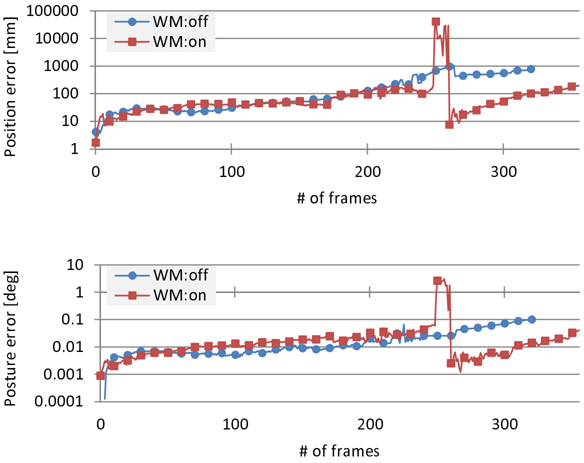
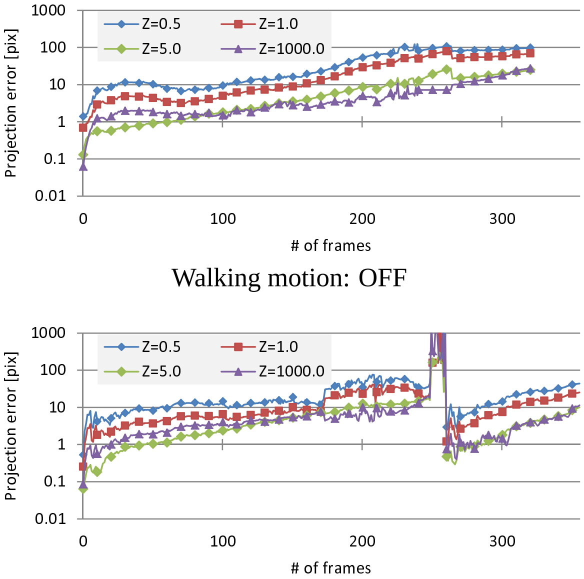


Figure 4. Position and posture errors



Walking motion: OFF

Walking motion: ON

Figure 5. Average projection errors

Acknowledgements

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

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.

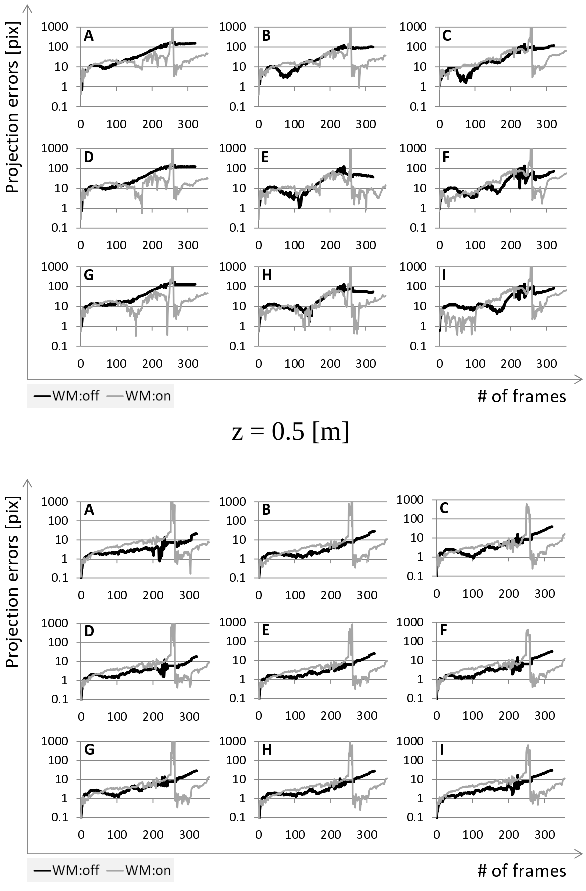
[8] TrakMark WEB
http://trakmark.net/

[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] T. Ishikawa, et al: “In-Situ 3D Indoor Modeler with a Camera and Self-Contained Sensors”, In Proc. of HCII2009, pp.454-464, 2009.

[11] 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.

[12] ISO (the International Organization for Standardization)/JTC1/SC24: http://www.iso.org/iso
/iso_technical_committee?commid=45252



z = 0.5 [m]

z = 1000 [m]

Figure 6. Projection errors of virtual objects