A Study of Camera Tracking Evaluation on TrakMark Data-Set

Masayuki Hayashi*
University of Tsukuba

Itaru Kitahara †
University of Tsukuba

Yoshinari Kameda ‡
University of Tsukuba

Yuichi Ohta §
University of Tsukuba

ABSTRACT

In this paper, we show a typical evaluation procedure of vision-based camera tracking using TrakMark [1] benchmark data-set. TrakMark data-set provides many image sequences with internal/external camera parameters, but evaluation criteria have not been defined yet. We discuss evaluation procedure of a camera calibration method with the TrakMark data-set; we take up Parallel Tracking and Mapping (PATM) [2] as an example because it is one of the major camera tracking techniques and available in augmented and mixed reality. We tested four image sequences of the TrakMark data-set. For qualitative evaluation, we visualize the camera path (3D trajectory of the camera position) to compare the result by PTAM and the ground truth given by TrakMark. For quantitative evaluation, we calculate the transform matrix from world coordinate system of the PTAM to the world coordinate system of the TrakMark. We build our program execution environment of the TrakMark on a USB-bootable Linux to make people easily conduct a further experiment and we are ready to distribute the environment.

KEYWORDS: TrakMark, Camera tracking, Visual SLAM, Trajectory, Evaluation

INDEX TERMS: I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—*Tracking*; H.5.1 [**Information Interfaces And Presentation**]: Multimedia Information Systems—*Artificial, augmented, and virtual realities*

1 INTRODUCTION

Vision-based camera tracking technique is one of the most important techniques in the field of augmented reality (AR) and mixed reality (MR). Although various approaches have been proposed for camera tracking technique, it is still difficult to compare the tracking performance of proposed algorithms since the evaluation method and data-set for that are not standardized yet. TrakMark working group make an effort to establish a standardized benchmark data-set [1]. They have already released eight packages that contain many kinds of image sequences, but evaluation criteria are not defined.

In this paper, we propose a typical evaluation procedure of vision-based camera tracking technique using TrakMark benchmark data-set. We take up parallel tracking and mapping (PTAM) [2] developed by Klein et al. as an example of camera tracking technique. We discuss how to evaluate PTAM on the TrakMark benchmark data-set, that requires initialization of 2-view stereo algorithm. Among eight packages of the TrakMark, we choose four packages including image sequences of outdoor and computer-generated scenes shown in Figure 1.

For qualitative evaluation, we visualize 3D trajectories of the estimated camera position and the corresponding camera position of the grand truth given by the TrakMark. This visualization is useful to grasp the overview of the estimated result.

To conduct a quantitative evaluation, we have to use the world coordinate system of the grand truth instead of PTAM's ad-hoc world coordinate system. We present a way to compute the transform matrix and evaluate the position and the orientation errors.

We build our program execution environment of the TrakMark so that the modified PTAM is available on USB-bootable Linux. This execution environment contains all the necessary dependent libraries and binary executables with source codes. It allows people to immediately conduct a further experiment on various environments since it can be booted from USB device.

Figure 1. Example images of the TrakMark data-set. We choose two computer-generated sequences (left-side) and outdoor sequences (right-side) for our evaluation.

2 PARALLEL TRACKING AND MAPPING ON TRAKMARK DATA-SET

We downloaded the PTAM source code from [4] and added some codes to perform our evaluation. TrakMark data-set is available at [3]. After downloading an image sequence, its associated intrinsic parameters and grand truth of extrinsic parameters, we conducted the evaluation by the steps below.

1. Convert the intrinsic camera parameters of TrakMark to "camera.cfg" format of PTAM.
2. Select a part of the image sequence for the stereo initialization of PTAM.
3. Start PTAM, and make a stereo initialization.
4. Compute the camera pose for the rest of the image sequence.

* e-mail: mhayashi@image.iit.tsukuba.ac.jp
† e-mail: kitahara@iit.tsukuba.ac.jp
‡ e-mail: kameda@iit.tsukuba.ac.jp
§ e-mail: ohta@acm.org

5. Compute the scaled Euclidean transform of the world coordinate system between PTAM and the grand truth.
6. Output the result.

Step 5 is optional because the step cannot be conducted without the grand truth. This step is not required to conduct our qualitative evaluation with ad-hoc world coordinate system of the PTAM. We think qualitative evaluation is useful even if this step is missed because we can directly see the shape of the trajectory of the estimated camera position and given grand truth camera position.

2.1 Intrinsic camera parameters

PTAM assumes that intrinsic camera parameters of normalized focal length (f_x, f_y), normalized principal point (u_0, v_0) and distortion parameter (ω) of Field of View model [6] are given. These parameters are saved in following format at "camera.cfg" file.

$$\text{Camera.Parameters} = [\ f_x\ f_y\ u_0\ v_0\ \omega\ ]$$

Distortion parameter ω corresponds with κ_1 of 4th order radial model [7]. ω is given in "Film Studio package" and "NAIST Campus package" of TrakMark data-set.

2.2 Stereo initialization

PTAM requires 2-view stereo initialization to make an initial 3D map of the scene. Since the tracking performance of PTAM depends on this initialization, we need to carefully select the images for the initialization of each image sequence to satisfy the following conditions.

1. The camera translates as slow as possible.
2. The camera should not rotate.
3. The initialization should be done as short as possible.

2.3 World coordinate system conversion

PTAM estimates the camera extrinsic parameters based on the initial map of the scene. Hence the world coordinate system of the initial map does not always correspond to the world coordinate system given by the grand truth. To perform a quantitative evaluation, we have to transform the world coordinate system of the initial map to that of the grand truth. We use scaled Euclidean transform which is computed the scaling factor and the Euclidean transform independently.

2.3.1 Euclidean transform

Assume that the stereo initialization has finished at *o*-th image of the sequence. The camera coordinates of PTAM and the grand truth are consistent at the moment. The scaled Euclidean transform $\mathbf{T}$ from the world coordinate system of PTAM to the world coordinate system of the ground truth can be denoted by Eq. (1).

$$\mathbf{T} = \mathbf{C}_{\mathrm{Go}}^{-1} \cdot \mathbf{S} \cdot \mathbf{C}_{\mathrm{Po}} \tag{1}$$

where $\mathbf{C}_{\mathrm{Po}}$ is the 4 by 4 matrix of extrinsic parameters of PTAM right after the stereo initialization, $\mathbf{C}_{\mathrm{Go}}$ is the corresponding ground truth and $\mathbf{S}$ is the uniform scaling matrix of which the scaling factor *s* is estimated in section 2.3.2.

We denote the 4 by 4 matrix of extrinsic parameters of PTAM and the grand truth at *i*-th image by $\mathbf{C}_{\mathrm{Pi}}$ and $\mathbf{C}_{\mathrm{Gi}}$. We transform $\mathbf{C}_{\mathrm{Pi}}$ that is related to the world coordinate system of PTAM to $\mathbf{C}_{\mathrm{Ei}}$ that is 4 by 4 extrinsic parameters related to the world coordinate system of the grand truth by Eq. (2). Then we compare $\mathbf{C}_{\mathrm{Ei}}$ with $\mathbf{C}_{\mathrm{Gi}}$.

$$\mathbf{C}_{\mathrm{Ei}} = \mathbf{C}_{\mathrm{Pi}} \cdot \mathbf{T}^{-1} \tag{2}$$

2.3.2 Scaling factor

We compute the uniform scaling matrix $\mathbf{S}$ through all images of the sequence and the grand truth. We denote PTAM estimated camera position at *i*-th image by $\boldsymbol{p}_{\mathrm{i}}$ and the correspondence of the grand truth by $\boldsymbol{r}_{\mathrm{i}}$. Euclidean distances from the camera position at *o*-th image $d_G(i)$ and $d_P(\mathrm{i})$ are given by Eq. (3).

$$\begin{aligned} d_G(i) &= ||\boldsymbol{r}_{\mathrm{i}} - \boldsymbol{r}_{\mathrm{o}}|| \\ d_P(i) &= ||\boldsymbol{p}_{\mathrm{i}} - \boldsymbol{p}_{\mathrm{o}}|| \end{aligned} \tag{3}$$

When the distance $d_G(i)$ becomes maximum at *m*-th image, we get the scaling factor *s* of $\mathbf{S}$ as a ratio of the $d_G(m)$ and $d_P(m)$.

$$\begin{aligned} m &= \arg\max_i(d_G(i)) \\ s &= d_{\mathrm{G}}(m)/d_P(m) \end{aligned} \tag{4}$$

Another method to compute the scaling factor is to give a correct baseline to the 2-view stereo initialization using the grand truth. Although we have tried the method, it causes more tracking errors.

3 Qualitative Evaluation

Using PTAM, we obtain an estimation of the 6-DOF camera pose relative to a world coordinate system from an image sequence captured by a single camera. Figure 1 shows the result of an image sequence "Translation + Panning" in Conference Venue Package 01 performed by PTAM. We can see the quality of augmentation by watching the 2D re-projection of the 3D feature points of the map and eyeball virtual object. To compare the estimated camera position and the corresponding ground truth provided by TrakMark, we visualize the 3D trajectory.

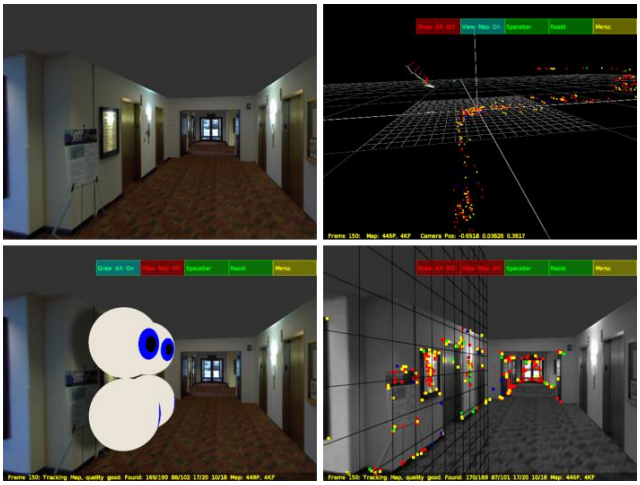

Figure 2. Example of PTAM on "Translation + Panning" sequence in Conference Venue Package 01 data-set. Upper left: source image. Lower left: eyeball virtual object. Upper right: 3D feature points of the map. Lower right: 2D re-projection of the map.

3.1 3D trajectory of the camera position

To see an accuracy of the camera position, we visualize the 3D trajectory. Figure 3 shows a result which does not match the world coordinate system of the PTAM and the grand truth. This unaligned visualization is still useful because we can intuitively see the difference of the shape of the trajectory of PTAM result and the grand truth.

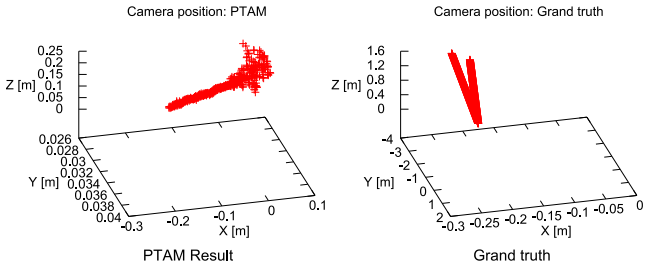

Figure 3. 3D trajectories of "Parallel translation" sequence in Conference Venue Package 01 data-set without world coordinate system conversion.

4 Quantitative Evaluation

To perform a quantitative evaluation, we converted the world coordinate system of the estimated extrinsic parameters as explained section 2.3. Figure 4 shows the 3D trajectories after the world coordinate system conversion. Because the two trajectories are aligned, now we are able to evaluate the accuracy of the camera position using Euclidean difference.

To evaluate the estimation accuracy of camera orientation, we take up an Euler angle between the estimation and the grand truth. The pseudo code we used to extract the Euler angle $\boldsymbol{v}$ [rad] from a 3 by 3 rotation matrix **R** are follows. The order of the angles is Z-X-Y.

```
If  R[3, 2] = 1
{
        v[X] =  π /2
        v[Y] = 0
        v[Z] = atan( R[2, 1] / R[1, 1] )
}Else If  R[3, 2] = -1
{
        v[X] = - π /2
        v[Y] = 0
        v[Z] = atan( R[2, 1] / R[1, 1] )
}Else
{
        v[X] = asin( R[3, 2] )
        v[Y] = atan2( -R[3, 1] / R[3,3] )
        v[Z] = atan2( -R[1, 2] / R[2, 2] )

}
```

Figure 5 shows an example of extracted Euler angles from the extrinsic parameters of PTAM estimation and the grand truth. We see that the angles are aligned accurately.

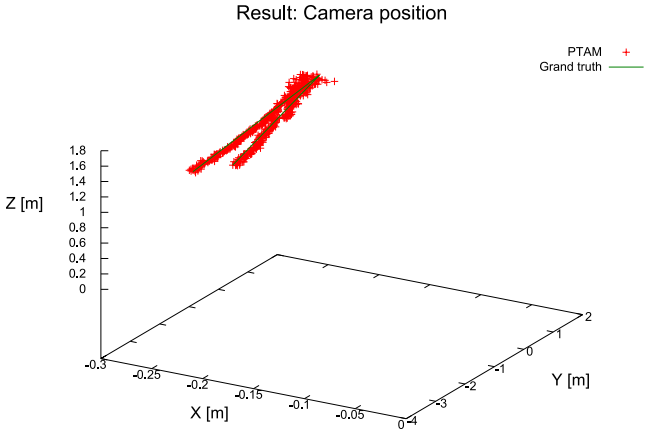

Figure 4. 3D trajectories of "Parallel translation" sequence in Conference Venue Package 01 data-set with world coordinate system conversion.

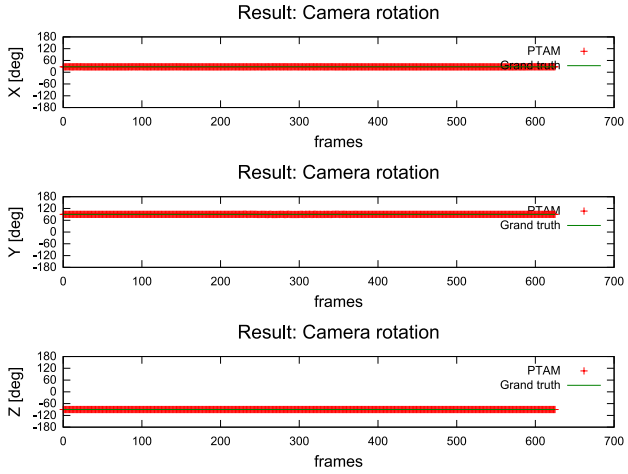

Figure 5. Extracted Euler angles from the extrinsic parameters of PTAM and the grand truth on "Parallel translation" sequence in Conference Venue Package 01 data-set.

5 Results

We have evaluated the PTAM using four packages including image sequences shown in Figure 1.

5.1 Conference Venue Package 01

This package contains three computer-generated image sequences: "Parallel translation", "Translation + Panning" and "Translation + Paninng + Tilting".

Figure 4 and 5 shows the result of "Parallel translation" sequence initialized with 1-40th images and Figure 7 shows the result of the "Translation + Panning" sequence initialized with 1-50th images and "Translation + Paninng + Tilting" sequence initialized with 90-140th images. The camera moves slowly in the sequences, therefore the camera tracking was successful for almost all frames in the sequences.

5.2 Nursing Home Package 01

This package contains two computer-generated image sequences "A" and "B" of a nursing home. Because the camera moves quickly, we shorten the interval of adding a new key-frame to the feature map of PTAM to 10 frames (20 frames by the default). Note that a practical limit of PTAM map size is around 6000

points and 150 key-frames because global bundle adjustment of the mapping cannot converge in a short time and it will be always aborted [2]. Therefore, the key-frame interval must not be shortened too much and 10 frames seem to be adequate for this sequence.

Figure 8 shows the result of the two sequences. The camera tracking was not succeeded at latter part of the sequences A (initialized with 1-31th images) and most of the sequence B (initialized with 20-40th images) because the camera moves too fast and hence the key-frame were not added properly.

5.3 NAIST Campus Package 01/02

These packages contain several image sequences captured by a monocular camera and an omnidirectional camera in outdoor environment.

Figure 9 shows the result of two monocular camera sequences; "mono" in the package 01 (initialized with 1-10th images) and "Sequence 00" in the package 02 (initialized with 1-7th images). The camera tracking failed for most of the images because we could not find good images for the stereo initialization which satisfy the conditions in section 2.2. Moreover, the camera moves too fast to add a new key-frame to the map.

5.4 Discussion

There are two major issues in our evaluation procedure of PTAM; the stereo initialization and world coordinate conversion.

With poor stereo initialization, poor feature points map will be produced and the camera tracking will be failed at most of the images after the initialization. We choose the images for the initialization empirically but we need sophisticated method.

To align the two camera trajectories of PTAM estimation and the grand truth, RMS error in geometry domain might be alternative choice.

6 USB-bootable Execution Environment

It is important to invite many researchers to the discussion to make the better evaluation criteria on TrakMark data-set. To have a detail review of our evaluation procedure, we decided to distribute our actual evaluation procedure[1].

Since our TrakMark program execution environment requires PTAM source code [4], its dependent libraries, patches of our modification, OpenCV [8], and TrakMark data-set, we build the environment on an USB-bootable Linux instead of only releasing source code of our modification so as to avoid set-up errors on building up the environment. Our environment is built on USB-bootable Linux (Ubuntu Linux 10.04 LTS) running on persistent mode [5]. We name our distribution package "Casper cartridge".

As shown in Figure 6, we package all of the software including operating system and device drivers into a USB memory. This package runs directly on various hardware including both desktop and laptop computers without any software installation to its original HDD. All software required for the PTAM evaluation procedure introduced in this paper has been installed on the environment. Furthermore, people can start extensive experiment of TrakMark and PTAM right after they boot the system.

Our program execution environment has two virtual file-systems in a physical file-system of the USB memory such as FAT32. The first is "squashfs" that is read-only file-system for USB-bootable Linux. This file-system is provided by Linux distributers. The other is "persistent file" that is read/write file-system unified to the squashfs transparently using UnionFS [9]. We install the all software required for our environment into this file-system. It is easy to back up the whole environment just by copying the persistent file because the persistent file is just a file when the USB memory is plugged to other PC (the filename is casper-rw).

The modified PTAM is able to load the image sequence from other storage device such as Blu-ray Disc since the TrakMark provides large data-set in BD media.

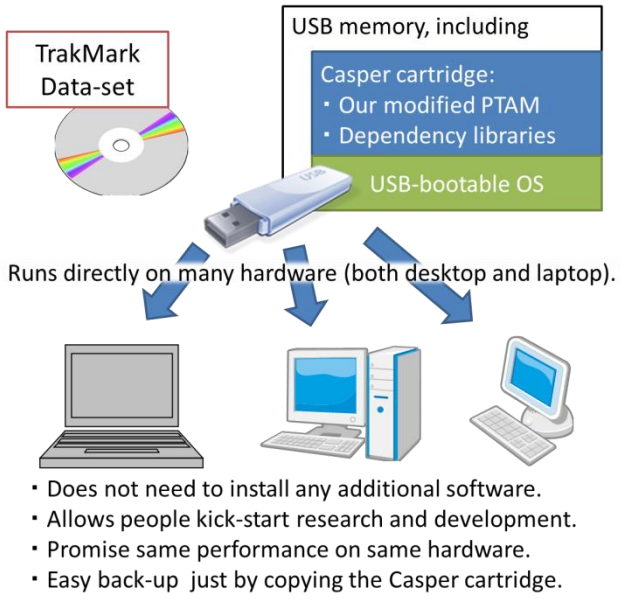

Figure 6. USB-bootable environment using "Casper cartridge". All software needs to run our modified PTAM is installed into a USB memory. Image sequence of the TrakMark data-set will be loaded from other storage devices such as Blu-ray Disc.

7 Conclusion

We have proposed a typical evaluation procedure of vision-based camera tracking using TrakMark data-set. We take up parallel tracking and mapping (PTAM) as an example of camera tracking technique. We choose four packages including image sequences of outdoor and computer-generated scenes. To qualitative evaluation, we visualize the 3D trajectories of the camera position that is useful to grasp the overview of the estimated result. We present a way to calculate the transform matrix to align the ad-hoc world coordinate of PTAM to the world coordinate of the grand truth.

To have a detail review of our evaluation procedure, we built our TrakMark execution environment on USB-bootable Linux and we are ready to distribute the environment. It makes easier to conduct a further experiment for other researchers since it runs on various hardware without any installation of its original HDD.

References

[1] H. Tamura, H. Kato and TrakMark Working Grougp, "Proposal of International Voluntary Activities on Establishing Benchmark Test Schemes for AR/MR Geometric Registration and Tracking Methods, Science and Technology," Proceedings of IEEE 2009 International Symposium on Mixed and Augmented Reality, pp. 233 - 236, 2009.

[2] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," Proceedings of IEEE 2007 International Symposium on Mixed and Augmented Reality (ISMAR), pp.1-10, 2007.

[3] TrakMark web site. http://trakmark.net/

[4] Parallel Tracking and Mapping for Small AR Workspaces - Source Code. http://www.robots.ox.ac.uk/~gk/PTAM/

[1] http://www.kameda-lab.org/casper

[5] Live USB Pendrive Persistent. https://wiki.ubuntu.com/LiveUsbPendrivePersistent
[6] F. Devernay and O. D. Faugeras, “Straight lines have to be straight,” Machine Vision and Applications, 13(1), pp.14-24, 2001.
[7] R. I. Hartley and A. Zisserman, “Multiple View Geometry in Computer Vision, 2nd Ed,” Cambridge: CUP, 2003.
[8] OpenCV. http://opencv.willowgarage.com/wiki/
[9] D. Quigley, J. Sipek, C. P. Wright and E. Zadok, "UnionFS: User- and Community-Oriented Development of a Unification Filesystem," Proceedings of the 2006 Linux Symposium, pp.349-362, 2006.

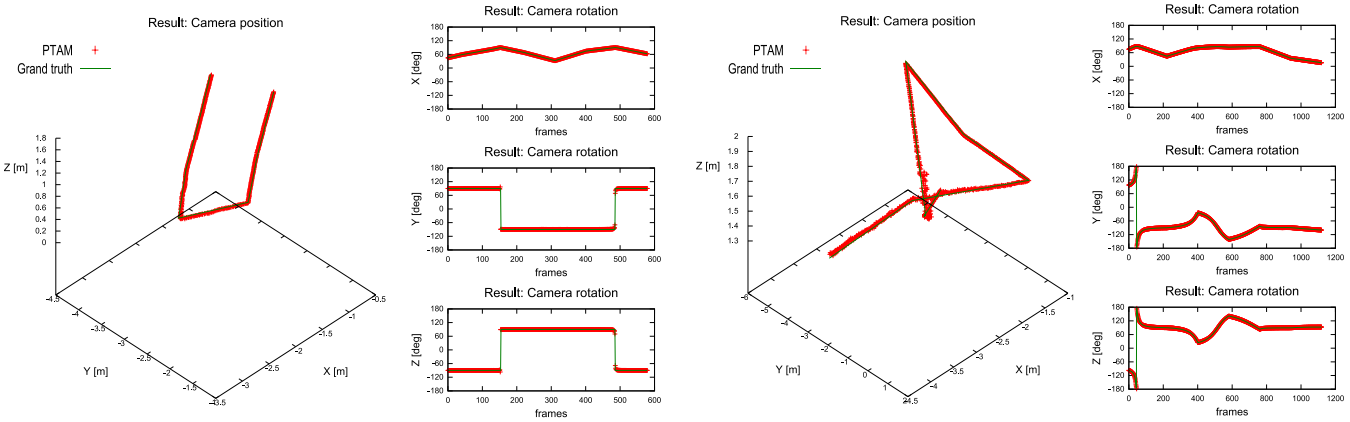

Figure 7. Result of image sequences in Conference Venue Package 01 data-set. Left two plots: “Translation + Panning”. Right two plots: “Translation + Panning + Tilting”.

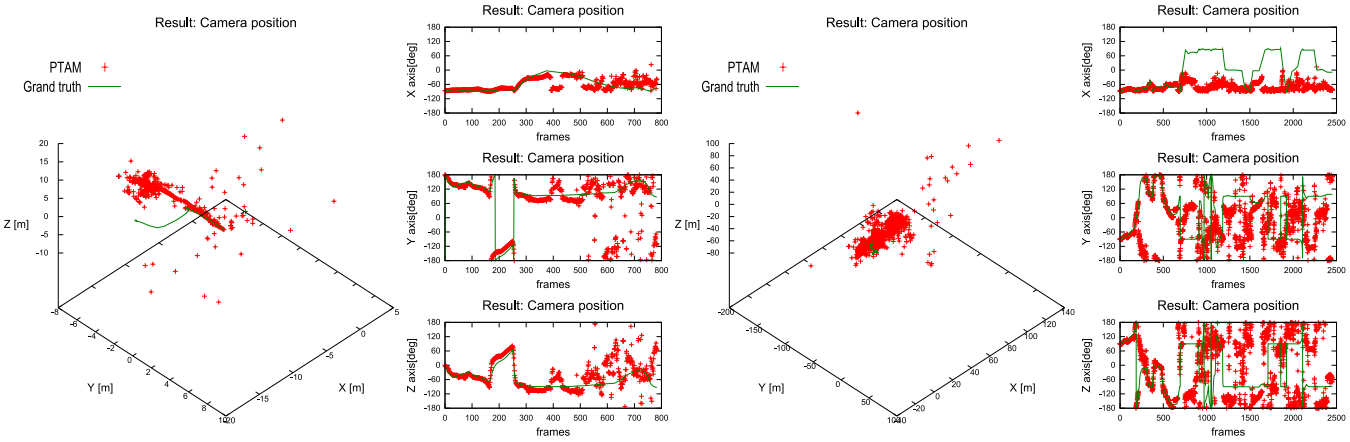

Figure 8. Result of image sequences in Nursing Home Package 01 data-set. Left two plots: sequence A. Right two plots: sequence B.

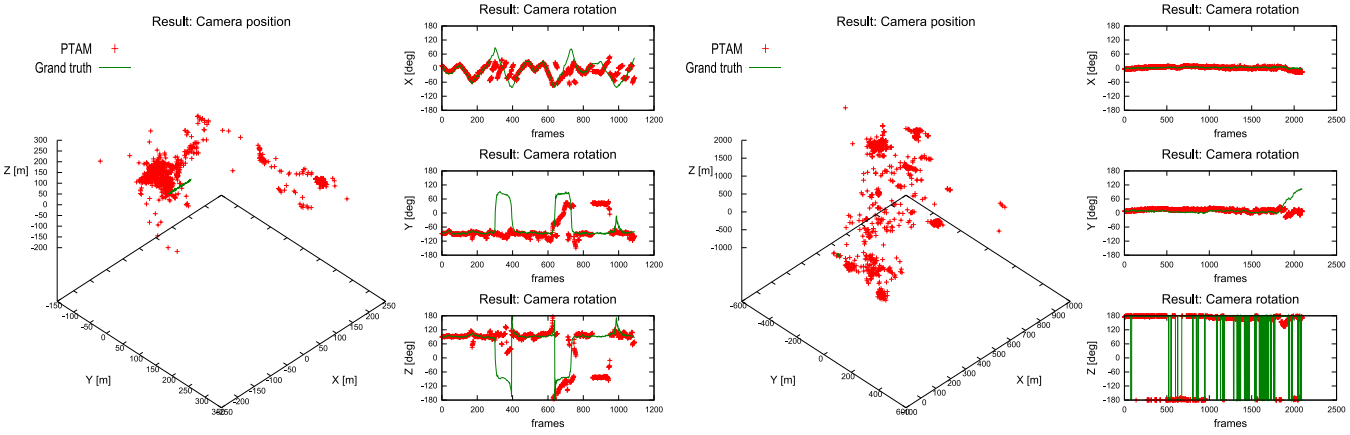

Figure 9. Result of image sequences in NAIST Campus Package 01 and 02 data-set. Left two plots: “mono” sequence of the package 01. Right two plots: “Sequence 00” in the package 02.