ROBOT POSE ESTIMATION: A VERTICAL STEREO PAIR VERSUS A HORIZONTAL ONE

Mohammad Ehab Ragab [1] and Kin Hong Wong[2]

[1]Informatics Department, The Electronics Research Institute, Giza, Egypt
[2]Computer Science and Engineering Department, Shatin, Hong Kong

ABSTRACT

In this paper, we study the effect of the layout of multiple cameras placed on top of an autonomous mobile robot. The idea is to study the effect of camera layout on the accuracy of estimated pose parameters. Particularly, we compare the performance of a vertical-stereo-pair put on the robot at the axis of rotation to that of a horizontal-stereo-pair. The motivation behind this comparison is that the robot rotation causes only a change of orientation to the cameras on the axis of rotation. On the other hand, off-axis cameras encounter additional translation beside the change of orientation. In this work, we show that for a stereo pair encountering sequences of large rotations, at least a reference camera should be put on the axis of rotation. Otherwise, the obtained translations have to be corrected based on the location of the rotation axis. This finding will help robot designers to develop vision systems that are capable of obtaining accurate pose for navigation control. An extensive set of simulations and real experiments have been carried out to investigate the performance of the studied camera layouts encountering different motion patterns. As the problem at hand is a real-time application, the extended Kalman filter (EKF) is used as a recursive estimator.

KEYWORDS

Pose estimation, robot navigation, multiple cameras, stereo, EKF

1. INTRODUCTION

Robot navigation, self-driven vehicles, and man-machine interaction are just a few examples of the countless applications that require solving the pose estimation problem (assessment of rotation and translation). This may be helpful for robots in understanding the environment and its objects which is a very important aspect in order for the robots to carry out their mission [1]. The Kalman filter (KF) is an optimal estimation algorithm for linear systems. However, in the field of computer vision, cameras are the typical sensors. Therefore, the linear assumption of the KF is violated by the perspective camera model. To deal with the nonlinearity of the image formation process, we use the extended Kalman filter (EKF) which resorts to calculating the derivatives (Jacobian) as a sort of linearization. In fact, the EKF is a suitable real-time estimator especially when the motion pattern is not of a chaotic nature (such as robot motion limited by motor capabilities and robot mass). To estimate robot pose, two back-to-back stereo pairs are used in [2], and the approach is studied further in [3] by comparing with subsets of cameras having different constructions of the EKF. Four non-overlapping cameras are motivated in [4] by the readiness for parallel processing deployment. The use of three cameras as a triple is compared to their use as two stereo pairs in [5].

Due to the difficulties facing the pose estimation (such as sensor noise, clutter, and occlusion), many researchers try to tackle the problem from more than one aspect. For example, in [6], a multiple sensor based robot localization system (consisting of optical encoders, an odometry model, a charge-coupled device (CCD) camera, and a laser range finder) is used. In [7] a mobile

DOI:10.5121/ijcsit.2015.7502

manipulator is located in a known environment using sensor fusion with the help of a particle filter. While in [8], a real-time system that enables a quadruped robot to maintain an accurate pose estimate fuses a stereo-camera sensor, inertial measurement units (IMU), and leg odometry with an EKF. Additionally in [9], a method for geo-localizing a vehicle in urban areas tackles the problem by fusing the Global Positioning System(GPS) receiver measurements, dead-reckoning sensors, pose prediction by 3D model and camera approach using Interacting Multiple Model - Unscented Kalman Filter (IMM-UKF). In [10], the mutual information between a textured 3D model of the city and a camera embedded on a vehicle is used to estimate its pose. Furthermore, in [11], a decentralized sensor fusion scheme is presented for pose estimation based on eye-to-hand and eye-in-hand cameras. Moreover to synchronize data, an approach is proposed in [12], for the online estimation of the time offset, between the camera and inertial sensors during EKF-based vision-aided inertial navigation.

Another way of facing the challenges of the problem is to use markers. For example in [6], the pose estimation is aided by a box and color blobs, and the system is validated using only the Microsoft Robotics Studio simulation environment. In [13], a fixed planar camera array is used to localize robots with color marks. While in [14], a methodology is used for trajectory tracking and obstacle avoidance of a car-like wheeled robot using two CCD cameras (fixed in location with pan-tilt capabilities). Two rectangular landmarks with green color and different size are used on the robot to help its localization. Besides, in [15], a stereo system is used in tracking a quadcopter with illuminated markers. In addition, in [16], a tiny single camera and an inertial measurement unit are used as two on-board sensors and two circles with different radii are chosen as the external markers. Also in [17], a localization system for cooperative multiple mobile robots is developed observing a set of known landmarks with the help of an omnidirectional camera atop each robot.

More on robot navigation can be found in [18] where a visual compass technique is proposed to determine the orientation of a robot using eight cameras, and in [19] which addresses the short baseline degeneration problem by using multiple essential matrices to regulate a non-holonomic mobile robot to the target pose. Furthermore, a broad introduction to estimating the ego-motion of an agent such as a robot is presented in [20] and in [21]. In fact, the work therein is a two-part tutorial which we will survey in the following lines for reasons mentioned below in this section. Instead of using “pose estimation”, the term “visual odometry” (VO) is used to denote the process of estimating the ego-motion of an agent (e.g., vehicle, human, and robot) using only the input of a single or multiple cameras attached to it. Three comparisons are held in [20]. The first compares the VO to the wheel odometry which depends on counting the number of wheel turns. Unlike the wheel odometry, the VO is not affected by wheel slip. Therefore, the VO provides more accurate pose estimation. The second compares the VO to the structure from motion problem (SFM). It is shown there that the VO is a special case of the SFM which (in addition to pose) seeks the 3D structure of the scene. Usually, the SFM needs refinement with an offline optimization such as the bundle adjustment. In contrast, the VO has to work in real time. The third comparison is conducted with the visual simultaneous localization and mapping (V-SLAM). It is indicated that the aim of the VO is the recursive estimation of robot pose. On the other hand, the target of the V-SLAM is obtaining a global and consistent estimate of the robot path. This includes constructing a map of the scene and detecting the robot return to previously visited locations (known as loop closure). Accordingly, the concern of the VO is the local coherence of the obtained path while the V-SLAM is concerned with the global coherence.

The conditions required by the VO are clarified in [20]. The VO detects the changes caused by the agent motion on the images captured by onboard cameras. Therefore, it needs an adequate illumination, a static scene with enough texture, and a degree of overlap among frames to obtain accurate estimates. The necessity of camera calibration to obtain the intrinsic camera parameters (such as the focal length), and the extrinsic parameters (such as the baseline between a stereo pair) is discussed. Depending on the specifications of feature correspondences, the VO is classified into three types: 2D to 2D, 3D to 3D, and 3D to 2D. Accordingly, some motion estimation approaches require using the triangulation to obtain the 3D structure of 2D feature correspondences of at least two frames. The triangulation is verified by the best possible intersection of back-projected rays from the 2D features in presence of noise and camera calibration errors. It is mentioned that some VO approaches have made use of the motion constraints to gain advantages with respect to speed and accuracy. For example, such constrains may determine the motion pattern to occur on a flat plane for vehicles. One of the problems encountering the VO is the drift. It is the accumulation of errors from frame to frame. When the drift becomes threatening to the accuracy, a sort of local optimization has to be utilized.

In [21], the feature selection methods are shown to belong to either one of two types. The first uses local search techniques such as correlation. An example of the features detected by this type is the corner which has high gradients in both the horizontal and vertical directions. The second detects features determining their descriptors in each frame then matches them among frames based on some similarity metric between such descriptors. A blob is an example of the features detected by the second type. The blob is neither an edge nor a corner but is a pattern that differs from its neighborhood in terms of color, texture, and intensity. The use of either type of feature selection depends on the application. For example, the scale-invariant feature transform (SIFT) is robust to changes in illumination, noise, minor changes in viewpoint, and partial occlusion. However, it automatically neglects corners which are abundant in man-made environments. Therefore, the choice of the appropriate feature detector should be thoroughly considered. With respect to tracking, it is shown that feature detection requires two steps. The first is applying a feature response function to the whole image. The second is applying the non-maxima suppression on the output of the first step. Then, the feature-matching step takes place to look for corresponding features in other images. The concept of mutual consistency check is a good measure of increasing the accuracy of matching (each feature of a matched pair is the preferred for the other). Another approach is to use an indexing structure, such as a hash table, to quickly search for features near a specific feature. Usually, wrong data associations (outliers) degrade the matching accuracy. So, the outlier rejection is the most delicate task in VO. Additionally, a sort of optimization called the pose graph optimization can enhance the accuracy of obtained pose parameters by minimizing a cost function. However, a nonlinear optimization scheme has to be used due to the presence of rotation which is a nonlinear part of the pose. There is an emphasis on the necessity of using the VO in global positioning system (GPS)-denied environments such as underwater. Such use is justified further by the texture-rich environment provided by the sea floor (which is ideal for computer vision approaches). The work in [21] is concluded by listing some publicly available code for building the VO systems.

There three reasons for surveying [20] and [21] above. The first is to introduce several concepts which will recur in the rest of this work. The second is to justify some of our approaches. For example, we use a corner-detector for features since corners are abundant in the indoor scene surrounding our robot. The third is to clarify a shortage of research regarding the effect of camera layout on the accuracy of pose obtained. This is one of the main reasons of conducting this work.
Here, we solve the pose estimation problem using only a multiple camera EKF approach in an unknown scene without using any other sensors or markers. The main contributions of this work are: suggesting a vertical stereo layout, comparing its use to the use of a horizontal stereo (and to the use of both) for robot pose estimation, and studying the effect of the reference camera position atop the robot on the accuracy of the pose obtained. The rest of this paper is organized as follows: the camera layout model, the EKF implementation, and the feature management are presented in section 2, the experimental results are shown in section 3, and the paper is concluded in section 4.

2. METHOD

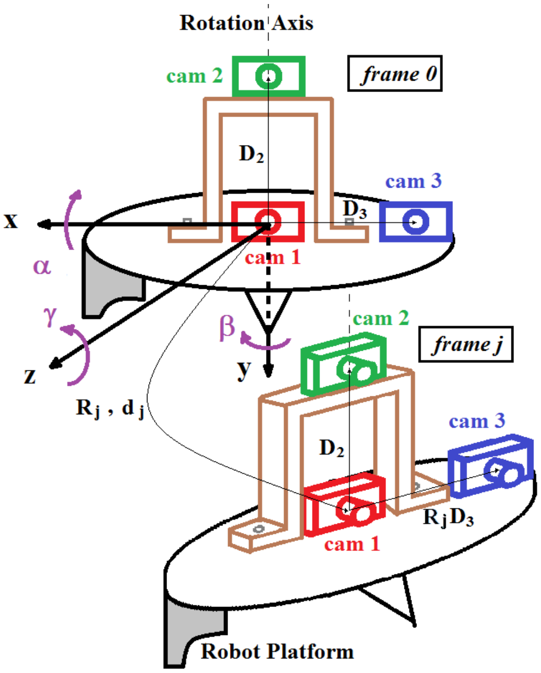

Figure 1. Vertical and horizontal stereo pairs, axes and angles.

In this section, we show the camera layout model, and the EKF implementation. Besides, we explain how to obtain features, which of them are fed to the EKF, and how to assure their validity.

2.1. Camera Layout Model

The multiple camera layout used is shown in Figure 1, where camera 1 and camera 2 form a vertical stereo pair. In the same time, a horizontal pair is formed by camera 1 and camera 3. The coordinate system used is a right-handed with the x and y axes pointing to the increase of the image coordinates in pixels (better seen from behind; and in this case the y axis is directed towards the floor). The rotation is described by the Euler angles α, β, and γ around the x, y, and z axes respectively. The reason for using the Euler angles is that they are directly related to the robot pose, hence more capable of describing its anatomy.

Before the motion starts (i.e. at *frame 0*), camera 1 has its center at the origin of the reference coordinate system. Camera 2 has its center displaced from camera 1 by the vector D_2 with the y component only not equal to zero. On the other hand, camera 3 has its center displaced from the center of camera 1 by the vector D_3 whose x component only not equal to zero. The z axis is perpendicularly emerging from camera 1 center towards the scene. All cameras are aligned parallel to the x and y axes (with rotation matrices equal to the identity matrix).

During the motion, at any general frame (*frame j*), camera 1 is rotated by the rotation matrix, R_j, with its center translated by the vector d_j with respect to the reference coordinate system. Our task is to estimate the pose (d_j, and R_j), or equivalently to find its six parameters (translation components in direction of the coordinate axes: t_{xj}, t_{yj}, and t_{zj}), and (rotation angles around the coordinate axes: α_j, β_j, and γ_j). The camera coordinates for camera 1 is given by:

$$P_{ij} = g(R_j)\,(M_i - d_j) \tag{1}$$

where M_i is a (3×1) vector defining the 3D location of the feature i (seen by the camera) with respect to the reference coordinate system, and $g(R_j)$ is a function of the camera rotation [22]. The camera coordinates for camera 3 is given by:

$$P_{ij3} = g(R_j)\,(M_i - d_j - R_j\,D_3\,) \tag{2}$$

Similarly, the camera coordinates of camera 2 is given by:

$$P_{ij2} = g(R_j)\,(M_i - d_j - R_j\,D_2\,) \tag{3}$$

However, as shown in Figure 1, camera 2 is located on the robot axis of rotation; therefore equation (3) is equivalent to:

$$P_{ij2} = g(R_j)\,(M_i - d_j - D_2\,) \tag{4}$$

This means that the location of camera 2 center is affected only by the robot translation (the rotation and the translation are decoupled as for camera 1). Nevertheless, we use the complete form of equation (3) in the experiments to test our approach without any prior information about the motion pattern. Therefore, taking e.g. camera 3 as the reference one, the obtained pose (d_{j3}, and R_{j3}) can be mapped to that seen by a point on the rotation axis by the following two equations:

$$d_j = d_{j3} - R_{j3}\,D_3 + D_3 \tag{5}$$

$$R_j = R_{j3} \tag{6}$$

What equation (5) actually verifies is removing the part of translation caused by off-axis rotation, and axis transferring. Equation (6) states that a camera put on the rotation axis would sense the same rotation if put anywhere else on the same rigid robot platform. Theoretically speaking, only if the camera shrank to a dimensionless point on the rotation axis, it would not be affected by any rotation whatsoever.

The rotation matrix of the robot at frame j, R_j, is related to the rotation angles by:

$$R_j = \begin{bmatrix} \cos\beta_j \cos\gamma_j & \sin\alpha_j \sin\beta_j \cos\gamma_j - \cos\alpha_j\, sin\gamma_j & \cos\alpha_j \sin\beta_j \cos\gamma_j + \sin\alpha_j \sin\gamma_j \\ \cos\beta_j \sin\gamma_j & \sin\alpha_j \sin\beta_j \sin\gamma_j + \cos\alpha_j \cos\gamma_j & \cos\alpha_j \sin\beta_j \sin\gamma_j - \sin\alpha_j \cos\gamma_j \\ -\sin\beta_j & \sin\alpha_j \cos\beta_j & \cos\alpha_j \cos\beta_j \end{bmatrix} \tag{7}$$

2.2. EKF Implementation

The EKF is a recursive estimator whose state space vector at frame j, s_j, consists of the pose parameters and their derivatives (velocities) in the form:

$$s_j = \begin{bmatrix} t_{xj} & \dot{t}_{xj} & t_{yj} & \dot{t}_{yj} & t_{zj} & \dot{t}_{zj} & \alpha_j & \dot{\alpha}_j & \beta_j & \dot{\beta}_j & \gamma_j & \dot{\gamma}_j \end{bmatrix}^T \tag{8}$$

Where the superscript T transforms the row to a column vector.

Additionally, the EKF has two main equations. The first is the plant equation which relates the current state space vector s_j to the previous one $s_{j\text{-}1}$ and the plant noise n_j assumed to be Gaussian:

$$s_j = A\,s_{j-1} + n_j \tag{9}$$

Where A is a (12×12) matrix dominated by zeros. The main diagonal has each element equal to one. The odd rows has a τ (equal to the time step between frames) just to the right of the main diagonal. In this way, a uniform motion of constant speeds are assumed for the robot.

The second equation is the measurement equation relating the 2D pixel locations of image features I_j, and the state measurement relation $h(s_j)$ (described below) assuming a Gaussian distribution for the measurement noise η_j:

$$I_j = h(s_j) + \eta_j \tag{10}$$

The state measurement relation is a function of each camera intrinsic parameters (such as the focal length, and the 2D center of projection), and the camera coordinates given in the equations (1) to (4) above. For each frame, the EKF predicts the state space vector based on the previous one, and updates it (enhancing the prediction) based on the measurements and the calculations of the Jacobian of the state measurement relation. More details about the EKF implementation can be found in [23-25]. In fact, the work in [23] by Broida et al. was one of the early attempts to bring the EKF to the field of computer vision which motivated a lot of the following related research.

2.3. Feature Management

The features mentioned above are small windows of pixels within the image frames. They are characterized by having a corner property (high intensity gradients in both directions). For each camera, the features are obtained and tracked using the Kanade-Lucas-Tomasi (KLT) feature tracker. For each stereo pair, they are matched based on a cross-correlation measure and the fundamental matrix encapsulating the epipolar constraints. In fact, this filters out the outliers.
Initially, corresponding matches of the calibrated stereo pairs are found. The locations of features in the 3D space are obtained using the triangulation. The features are tracked from frame to frame for individual cameras. The features violating the epipolar constraints are filtered out. The requirement of the EKF to have zero mean measurements is taken into consideration. Therefore, the features fed to the filter are chosen to be as evenly distributed as possible around the center of projection of each image. Accordingly, the set of features may vary from frame to frame. When the number of tracked features falls under a certain threshold, a new set of fresh features is obtained using the stereo matching as mentioned above. The choice of the threshold depends on the image size and the nature of the scene. For the (1600×1200) images taken in the ordinary lab scene used in this work, a threshold of 140 features is a suitable choice.

3. EXPERIMENTS AND RESULTS

3.1 Simulations

Three cameras forming two stereo pairs (vertical and horizontal) were put on a robot as shown in Figure 1.The robot moved with random translations (t_x, t_y, and t_z) and with random rotation angles (α, β, and γ) in the directions of and around the x, y, and z axes respectively. The coordinate system origin coincided with the center of the first camera at the motion start with the z axis perpendicular to the image plane. To have a right-hand coordinate system, the x axis originates towards the opposite direction of cam 3, as the y axis does with respect to cam 2. The translations were taken randomly from ±0.005 to ±0.015 meter, and the rotation angles were taken randomly from ±0.005 to ±0.02 radian. All cameras had a 6 mms focal length, and (640×480) resolution. The center-to-center distance for the horizontal stereo pair was 0.1 meter similar to the vertical stereo pair. A random noise was added to each feature point with a normal distribution of zero mean and a 0.5 pixel standard deviation. The motion took place inside a sphere whose radius was one meter and whose center was coinciding with the origin of the coordinate axes. The feature points were distributed randomly in a spherical shell located between that sphere and a smaller one with a radius of 2/3 meter. The total number of feature points, was 10,000. A sequence of 100 frames was taken simultaneously by each camera. Due to the motion randomness, the sequences were divided into a number of sections which contained ten frames each. Throughout each section new features were tracked. Using the multiple camera model described in section 2, we solved for the pose parameters utilizing pose EKFs for each stereo pair (v_{str}: vertical stereo, and h_{str}: horizontal). For the sake of completeness, we formed a triple of all cameras. To enlarge the scope of the simulations, we varied the motion patterns to have a pure rotation, a pure translation, and a mixture of both. Then we increased the ranges of translations and rotations for each step to be up to ±0.0225 meter, and ±0.03 radian respectively and ran the three motion patterns again. Table 1 shows the average of (500 runs per motion pattern) of absolute error (per frame) in the six pose parameters for the compared methods. All absolute errors are given in (meter/radian).

3.2 Real Experiments

Table 1. Average absolute error of pose values per frame for simulations. 'Large' indicates the motion patterns whose ranges of translation and rotation were increased, 'v_{str}' is the vertical stereo, 'h_{str}' is the horizontal, and 'triple' is both stereo pairs composed from the three cameras.

	Motion Pattern	t_x	t_y	t_z	α	β	γ
v_{str}	Mixed	.0102	.0030	.0026	.0029	.0099	.0012
h_{str}		.0103	.0036	.0029	.0036	.0101	.0013
triple		.0100	.0029	.0026	.0028	.0097	.0011
v_{str}	Pure Rotation	.0096	.0015	.0015	.0016	.0091	.0008
h_{str}		.0097	.0025	.0025	.0026	.0092	.0009
triple		.0096	.0016	.0018	.0017	.0091	.0007
v_{str}	Pure Translation	.0032	.0028	.0013	.0028	.0033	.0005
h_{str}		.0033	.0040	.0014	.0041	.0034	.0008
triple		.0030	.0030	.0013	.0031	.0031	.0004
v_{str}	Large Mixed	.0194	.0051	.0056	.0050	.0179	.0023
h_{str}		.0193	.0054	.0054	.0051	.0177	.0024
triple		.0187	.0048	.0056	.0046	.0173	.0021
v_{str}	Large Pure Rotation	.0193	.0030	.0028	.0031	.0180	.0019
h_{str}		.0195	.0033	.0032	.0035	.0189	.0018
triple		.0201	.0028	.0041	.0029	.0181	.0016
v_{str}	Large Pure Translation	.0040	.0039	.0027	.0038	.0040	.0006
h_{str}		.0044	.0048	.0028	.0048	.0043	.0008
triple		.0039	.0039	.0027	.0038	.0039	.0005

We carried out the real experiments using three calibrated Canon PowerShot G9 cameras with resolution (1600×1200). The cameras were put parallel atop the robot used with the baseline of each stereo pair equal to 14 cm. A sequence of more than 200 frames was taken simultaneously by each camera in an ordinary lab scene. The motion of the robot followed various patterns: a pure translation, a pure rotation, a mixed rotation and translation, a large pure rotation, and a curve. These motion patterns are shown graphically in the figures below. We did not capture each sequence twice using the same two cameras as a horizontal stereo pair then as a vertical stereo pair. Instead, we captured each sequence only once using three cameras simultaneously. The reasons for this are guaranteeing that both stereo pairs encounter exactly the same sequence and eliminating any possible difference due to the wheel-slipping effect. Besides, we can compare the performances of both stereo pairs to that of the triple. Moreover, we exchanged the cameras (within the predefined layout) from a sequence to another to eliminate the effect of any discrepancy in camera calibration (if any). Our robot maneuvers on a plane floor and the cameras are firmly fixed onto the platform. Therefore, only three pose parameters can really vary with the robot motion. They are the translations t_x and t_z, and the rotation angle β around the y axis. In fact, all the motion patterns investigated in the real experiments are obtained by varying one or more of these three pose parameters. The three other pose parameters (t_y, α, and γ) may vary slightly due to the vibrations caused by the robot suspension. However, for completion, we include all the six pose parameters in our figures. The robot and camera setup used in the experiments along with samples of the captured images are shown in Figure 2. The graphical results are shown in the rest of figures compared with the ground truth obtained from the computer steering the robot.

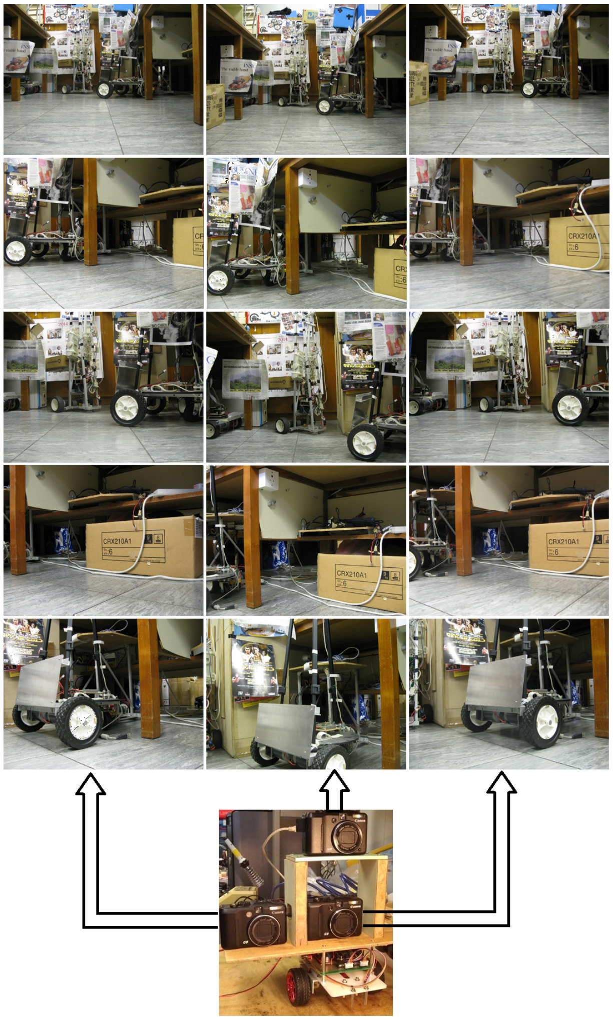

Figure 2. Rows show frames taken simultaneously, from top: pure translation, pure rotation, mixed, large pure rotation, and curve; below: camera setup and robot with arrows from a camera to its sample frames.

4. Discussions and Conclusions

As for the simulations, from Table 1, it is clear that as the range of steps of pose parameters increases, the average absolute errors increase. For example, the pattern "Large Mixed" has higher average absolute errors than the pattern "Mixed". As expected, using all the cameras (triple) verifies the less average absolute errors in most cases. Next to its performance and very close is the vertical stereo pair (v_{str}). The main reason for this is decoupling the rotation around the robot axis from the translation as shown in equation (4) above. This is emphasized by the lowest average absolute errors for t_x, t_z , and β for the pure translation patterns. In fact, β takes place around the y axis in the same plane containing t_x and t_z. Therefore, estimating any of them accurately, increase the possibility of estimating the other two parameters accurately as well.
In the following lines, we will survey the results of the real experiments. Figure 3 depicts the pure translation pattern obtained by increasing t_z linearly. The vertical stereo, v_{str}, the horizontal stereo, h_{str}, and the triple are close to each other and to the ground truth. There is a slight drift near the end of the sequence in the angle β. This may be caused by wheel-slipping or not precisely synchronized motors (even for a few frames). The translations t_x, and t_z are slightly affected due to the relation with β mentioned above.

Similarly, the pure rotation, the mixed rotation and translation, and the pure large rotation patterns have the compared sets of cameras close to each other and to the ground truth with a slight drift. Figure 4 portrays a linear decrease in angle β while Figure 5 shows β increasing linearly at a larger rate. In Figure 6, the translation t_z increases piecewise linearly while β varies around zero within ±0.1 radian.

When it comes to the curve pattern, the comparison becomes more obvious. As shown in Figure 7, the angle β increases at a large pace while both the translations t_x, and t_z follow nonlinear curves. The challenges encountered are varying all the three pose parameters available for a plane motion and having translations which are not uniform (speeds are not constant). This can be shown by taking the slopes along the nonlinear curves. The vertical stereo, v_{str}, is closer to the ground truth for the translations t_x, and t_z during most of the sequence. Even when some drift appears near the end of the sequence, its magnitude remains close to that encountered by the horizontal stereo. For all pose parameters, the triple could verify the best performance. In this case, the data fusion of the three cameras has paid off.

From now on, we investigate the effect of choosing a reference camera other than camera 1. In Figure 8, exchanging the reference camera within the horizontal stereo pair does not alter the pose obtained for the pure translation pattern. However, in the pure rotation pattern of Figure 9, a considerable translation component appears for t_x and even more for t_z of the exchanged stereo pair h_{str3-1}. The reason for this is that the reference camera (camera 3 in this case) encounters additional translation being displaced from the rotation axis. Although the detected translation is real for camera 3, it does not occur for the center parts of the robot. Moreover, the magnitude of the additional translation varies proportionally with the displacement. As seen in Figure 9, the rotation angles do not suffer from any alteration. However, the obtained translation can be mapped back to that seen by a reference camera on the robot axis of rotation using equation (5) above. As shown in Figure 9, the corrected pose, $h_{str3-1C}$ is close to the ground truth. Additionally, the pose obtained by exchanging the reference camera within the vertical stereo pair (camera 2 becoming the reference) does not suffer from any variation being on the robot axis of rotation as camera 1.

Examining the mixed rotation and translation pattern, in Figure 10, the exchange of the reference camera has no observed effect on the obtained pose parameters. This is caused by the small variation range of β within ±0.1 radian around zero which is close to the pure translation case.
The effect of exchanging the reference camera within the horizontal stereo pair (h_{str3-1}) is amplified for the curve pattern as shown in Figure 11. The influence on t_z is obvious throughout

the sequence while it is manifested for t_x with the increase in β. Using equation (5) to obtain the corrected pose ($h_{str3\text{-}1C}$) brings the translations back to that of ($h_{str1\text{-}3}$) with an obvious drift in t_z throughout most of the sequence. Again, exchanging the reference camera within the vertical stereo pair does not alter the estimated pose.

Comparing to the vertical stereo pair, we have noticed that the horizontal stereo pair has estimates for (t_y, α, and γ) generally closer to zero (as it should be). The reasons for this are embedded in equations (2), and (3). In equation (2), a term of the camera coordinates for camera 3 is a matrix-by-vector multiplication ($R_j\, D_3$). As D_3 has only an x component, the multiplication turns out to be a scaling of the first column of R_j which as seen from equation (7) is dominated by the angle β with a little effect from the angle γ. On the other hand in equation (3), the multiplication ($R_j\, D_2$) turns out to be a scaling of the second column of R_j with a far more influence from the angles α, and γ. In this way, using the horizontal stereo pair attenuates the fluctuations about zero of the estimates for (t_y, α, and γ) while the vertical stereo pair magnifies them. Nevertheless, the fluctuations noted remain small and the values for (t_y, α, and γ) are known to be zeros from the beginning for a robot moving on a plane floor.

This work proves that the camera layout is crucial for accurate robot pose estimation. Having a vertical stereo pair whose cameras' centers lie on the robot axis of rotation decouples the translation from the rotation. This is particularly useful for getting more accurate pose estimation when the motion patterns combine large rotations with nonlinear translations. The accuracy is emphasized when the robot motion takes place on a plane floor, and there are adequate features in the scene to track.

Putting the reference camera on the axis of rotation is important. When it comes to robot navigation, decoupling the translation from the rotation is not only more accurate but could be more energy efficient also (with fewer maneuvering instructions). A correction can be made to map the estimated translations to that encountered on the rotation axis. However, this requires knowing the displacement of the reference camera from the robot axis of rotation. Such displacement is readily available by stereo calibration when one of the cameras (whether it is the reference or not) is put on the axis of rotation. Therefore, the camera on the rotation axis should rather be taken as the reference.

Practically, a robot axis of rotation might change its location from a rotation to another. The cause may be unmatched or unsynchronized motors. Nevertheless, the most accurate location for the axis can be found by rotation calibration (before first use and whenever it is needed). The real challenge is determining this axis when the robot has several wheels with all-wheel drive or when it has articulated parts. Suggesting the most suitable camera layouts in these cases would be a topic for future research as well as making use of Big Data principles [26-29] in dealing with multiple cameras.

REFERENCES

[1] S. S. Hidayat, B. K. Kim, and K. Ohba, "An approach for robots to deal with objects", *International Journal of Computer Science & Information Technology (IJCSIT)*, Vol. 4, No. 1, pp 19-32. DOI : 10.5121/ijcsit.2012.4102 19

[2] M.E. Ragab, K.H. Wong, J.Z. Chen, and M.M.Y. Chang, "EKF based pose estimation using two back-to-back stereo pairs", *ICIP'07,* IEEE, pp. 137–140.

[3] M.E. Ragab, K.H. Wong, J.Z. Chen, and M.M.Y. Chang, "EKF pose estimation: how many filters and cameras to use?", *ICIP'08,* IEEE, pp. 245–248.

[4] M.E. Ragab, K.H. Wong, "Multiple nonoverlapping camera pose estimation", *ICIP'10,* IEEE, pp. 3253 - 3256.

[5] M.E. Ragab, K.H. Wong, "Three camera robot pose estimation: a triple versus two pairs", *Robotics Applications 2014,* IASTED, DOI: 10.2316/P.2014.817-018.

[6] Haoxiang Lang, Ying Wang, and Clarence W. de Silva, “Mobile robot localization and object pose estimation using optical encoder, vision and laser sensors”, *International Conference on Automation and Logistics 2008,* IEEE, pp 617-622.

[7] Andrea Gasparri, Stefano Panzieri, Federica Pascucci, and Giovanni Ulivi, “Pose recovery for a mobile manipulator using a particle filter”, *MED* 2006, IEEE, pp1-6.

[8] Jeremy Ma, Sara Susca, Max Bajracharya, and Larry Matthies, “Robust multi-sensor, day/night 6-dof pose estimation for a dynamic legged vehicle in GPS-denied environments”, *International Conference on Robotics and Automation 2012*, IEEE, pp 619-626.

[9] Maya Dawood, et al., “Harris, SIFT and SURF features comparison for vehicle localization based on virtual 3D model and camera”, *Image Processing Theory, Tools and Applications 2012,* IEEE, pp 307-312.

[10] Guillaume Caron, Amaury Dame, Eric Marchand, “Direct model based visual tracking and pose estimation using mutual information”, *Image and Vision Computing*, 32 (2014), pp 54–63.

[11] Akbar Assa, and Farrokh Janabi Sharifi, “Decentralized multi-camera fusion for robust and accurate pose estimation”, *AIM 2013,* IEEE/ASME, pp 1696-1701.

[12] Mingyang Li, and Anastasios I. Mourikis, “3-D motion estimation and online temporal calibration for camera-IMU systems”, *ICRA 2013,* IEEE, pp 5709-5716.

[13] Xuefeng Liang, and et al., “Multiple robots localization using large planar camera array for automated guided vehicle system”, *International Conference on Information and Automation 2008,* IEEE, pp 984-990.

[14] Chih-Lyang Hwang, and Chin-Yuan Shih, “A distributed active-vision network-space approach for the navigation of a car-like wheeled robot”, IEEE *Transactions on Industrial Electronics,* Vol. 56 No. 3, pp 846-855.

[15] Markus Achtelik, Tianguang Zhang, Kolja Kiihnlenz and Martin Buss, “Visual tracking and control of a quadcopter using a stereo camera system and inertial sensors”, *International Conference on Mechatronics and Automation 2009,* IEEE, pp 2863-2869.

[16] Tianguang Zhang, Ye Kang, Markus Achtelik, Kolja Kiihnlenz, and Martin Buss, “Autonomous hovering of a vision/IMU guided quadrotor”, *International Conference on Mechatronics and Automation 2009,* IEEE, pp 2870-2875.

[17] Shigekazu Hukui, Jie Luo, Takuma Daicho, Keigo Suenaga, and Keitaro Naruse, “Mutual localization of sensor node robots”, *ISAC 2010,* IEEE, pp 104-110.

[18] Anupa Sabnis, and Leena Vachhani, “A multiple camera based visual compass for a mobile robot in dynamic environment”, *CCA 2013,* IEEE, pp 611-616.

[19] Baoquan Li, Yongchun Fang, and Xuebo Zhang, “Essential-matrix-based visual servoing of nonholonomic mobile robots without short baseline degeneration”, *CCA 2013,* IEEE, pp 617-622.

[20] Davide Scaramuzza, and Friedrich Fraundorfer, “Visual odometry part I: the first 30 years and fundamentals”, *Robotics & Automation,* IEEE, (December 2011), pp 80-92.

[21] Friedrich Fraundorfer, and Davide Scaramuzza, “Visual odometry part II: matching, robustness, optimization, and applications”, *Robotics & Automation,* IEEE, (June 2012), pp 78-90.

[22] M. E. Ragab, K.H. Wong, “Rotation within camera projection matrix using Euler angles quaternions and angle-axes”, *International Journal of Robotics and Automation,* ACTA Press, Vol. 24, No. 4, pp 312-318.

[23] T.J. Broida, S. Chanrashekhar, and R. Chellappa, “Recursive 3-D motion estimation from a monocular image sequence”, *Transactions on Aerospace and Electronic Systems,* IEEE, Vol. 26, No. 4, pp 639–656.

[24] A. Azarbayejani, and A.P. Pentland, “Recursive estimation of motion, structure, and focal length”, *Transactions on Pattern Anal. Mach. Intell.,* IEEE, Vol. 17, No. 6, pp 562-575.

[25] V. Lippiello, B. Siciliano and L. Villani, “Position and orientation estimation based on Kalman filtering of stereo images”, *CCA 2001*, pp 702-707.

[26] S. Sharma, U. S. Tim, J. Wong, S. Gadia, and S. Sharma, “A brief review on leading big data models”, *Data Science Journal*, Vol. 13, pp138-157.

[27] S. Sharma, R. Shandilya, S. Patnaik, and A. Mahapatra, “Leading NoSQL models for handling big data: a brief review”, *International Journal of Business Information Systems*, Inderscience, 2015.

[28] S. Sharma, U. S. Tim; S. Gadia, J. Wong, R. Shandilya, and S. K. Peddoju, “Classification and comparison of NoSQL big data models”, *International Journal of Big Data Intelligence* (*IJBDI*), Vol. 2, No. 3, 2015.

[29] Sharma, S. (2015). Evolution of as-a-Service Era in Cloud. arXiv preprint arXiv:1507.00939.

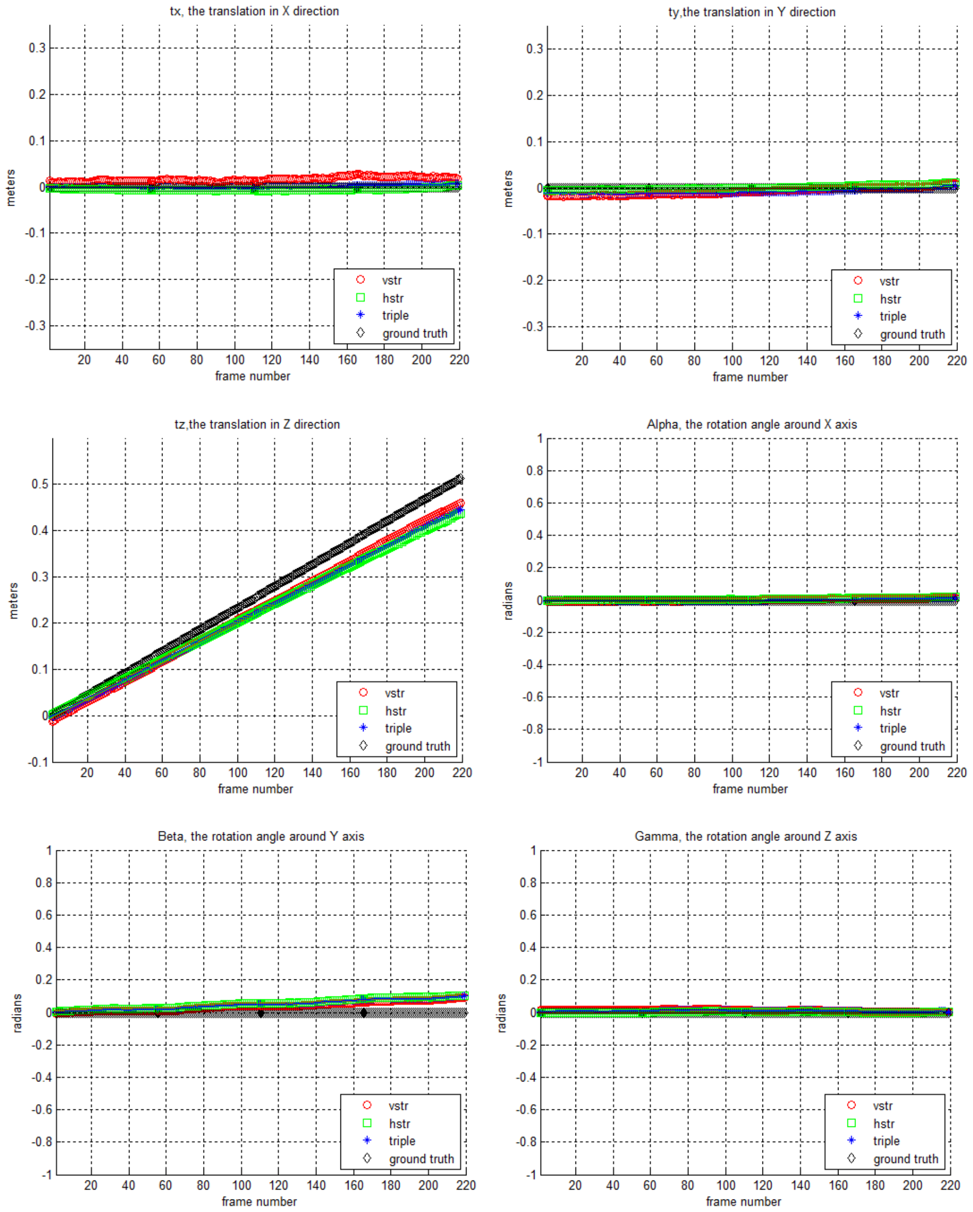

Figure 3. Results of real experiment (pure translation pattern), ‘v_{str}’ is the vertical stereo, ‘h_{str}’ is the horizontal stereo, and the ‘triple’ is the three cameras. All are close to the ground truth for this pattern.

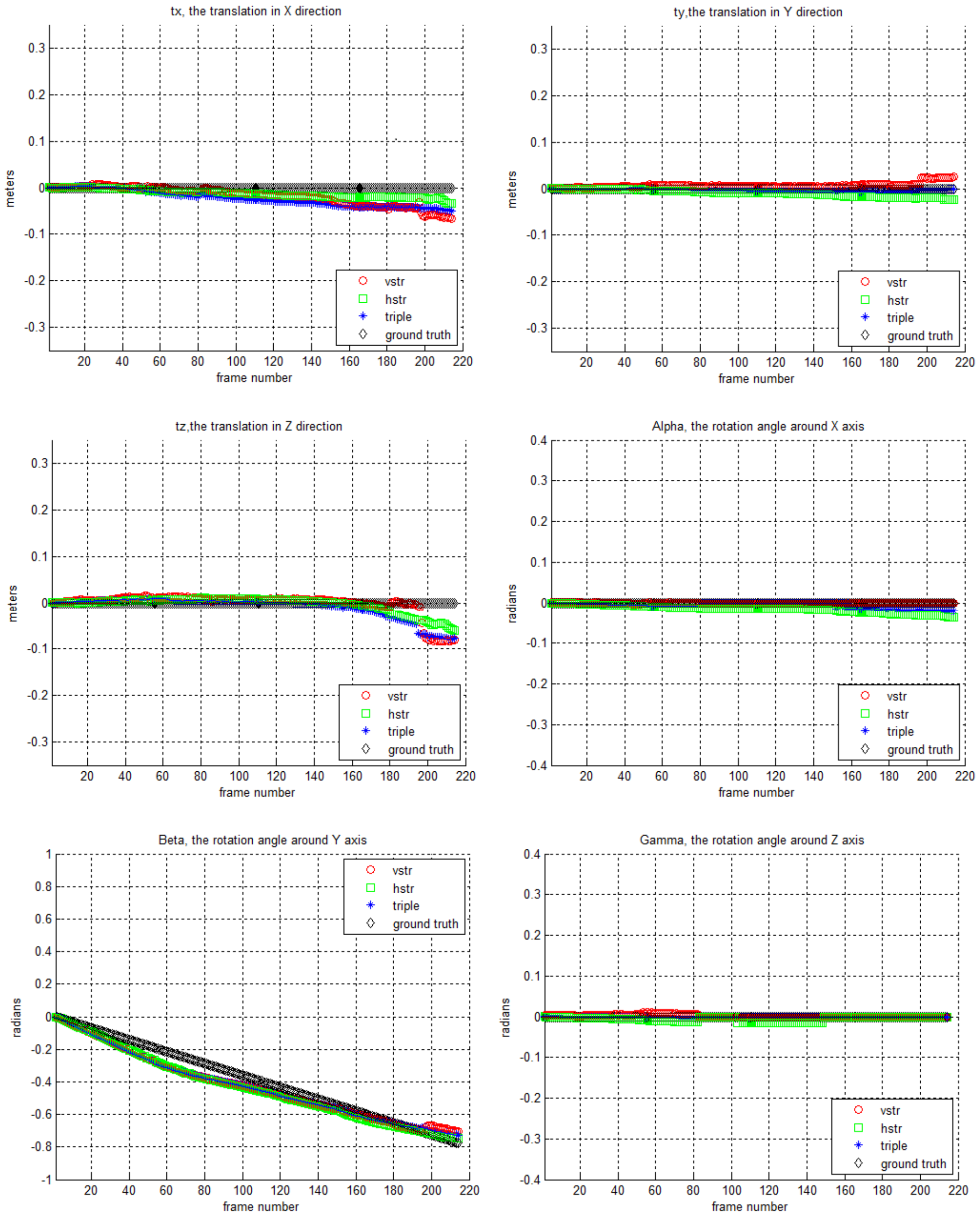

Figure 4. Results of real experiment (pure rotation pattern), ‘v_{str}’ is the vertical stereo, ‘h_{str}’ is the horizontal stereo, and the ‘triple’ is the three cameras. All are close to the ground truth for this pattern.

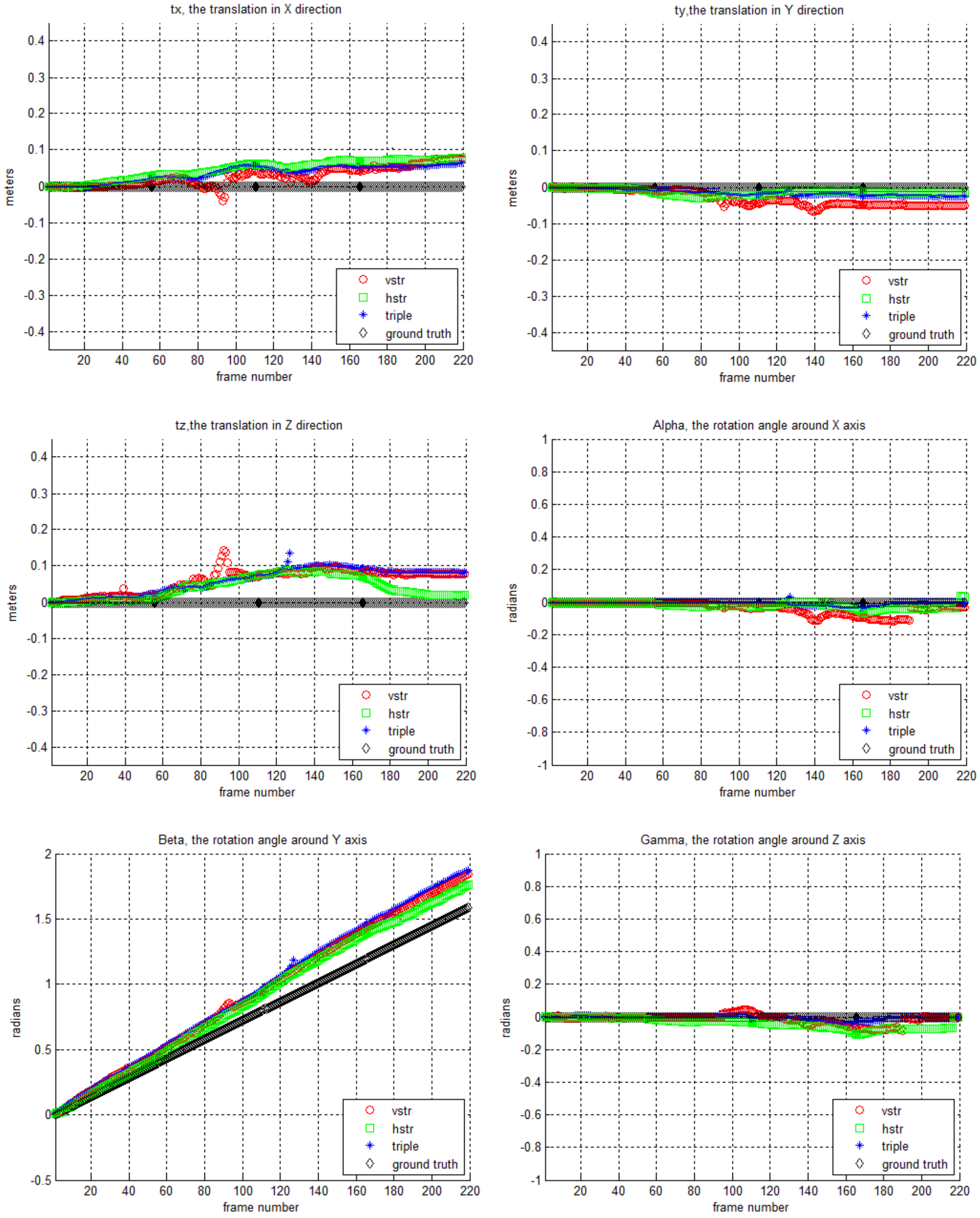

Figure 5. Results of real experiment (large pure rotation pattern), ‘v_{str}’ is the vertical stereo, ‘h_{str}’ is the horizontal stereo, and the ‘triple’ is the three cameras. Slight differences are noticed.

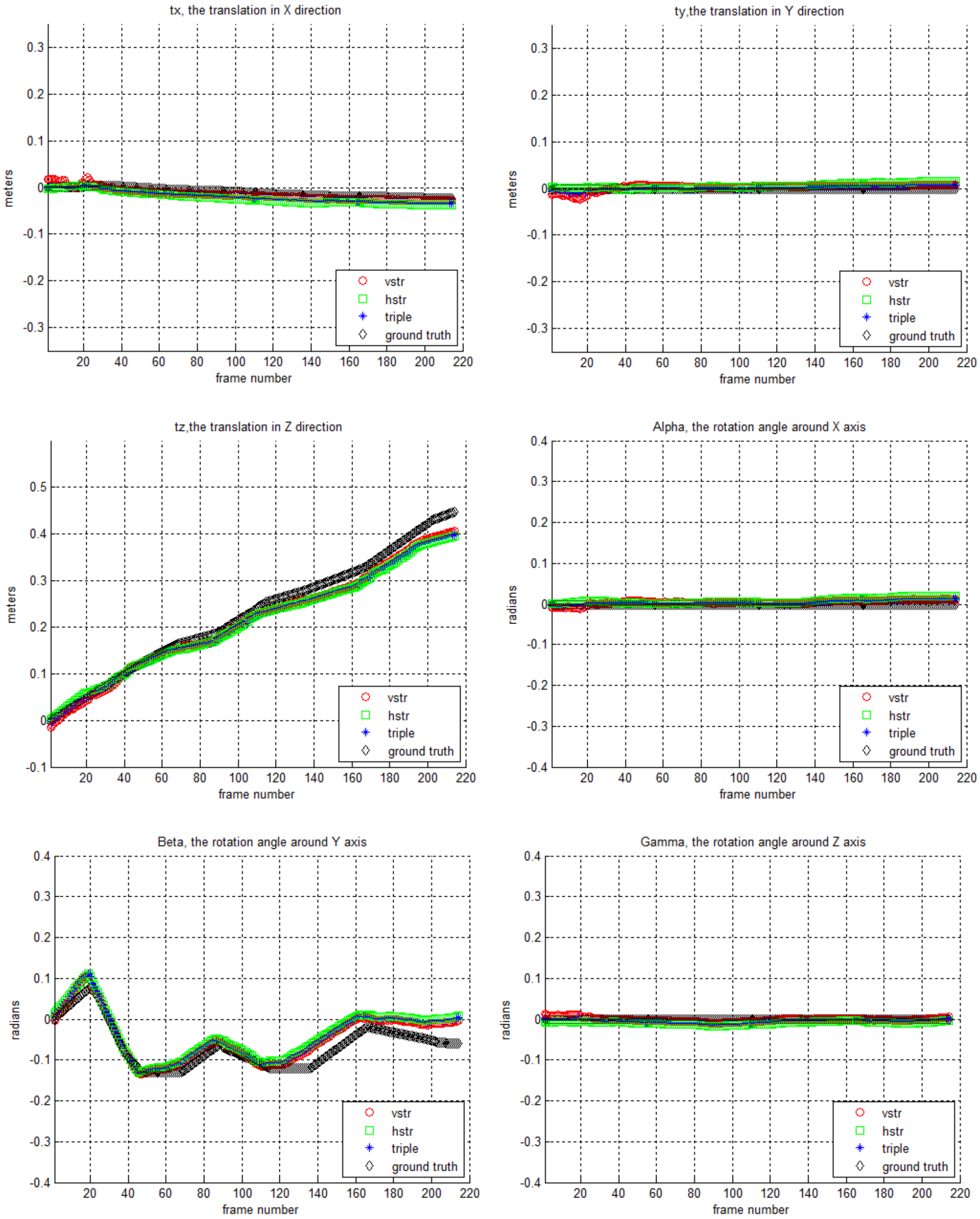

Figure 6. Results of real experiment (mixed pattern), ‘v_{str}’ is the vertical stereo, ‘h_{str}’ is the horizontal stereo, and the ‘triple’ is the three cameras. All are close to the ground truth for this pattern.

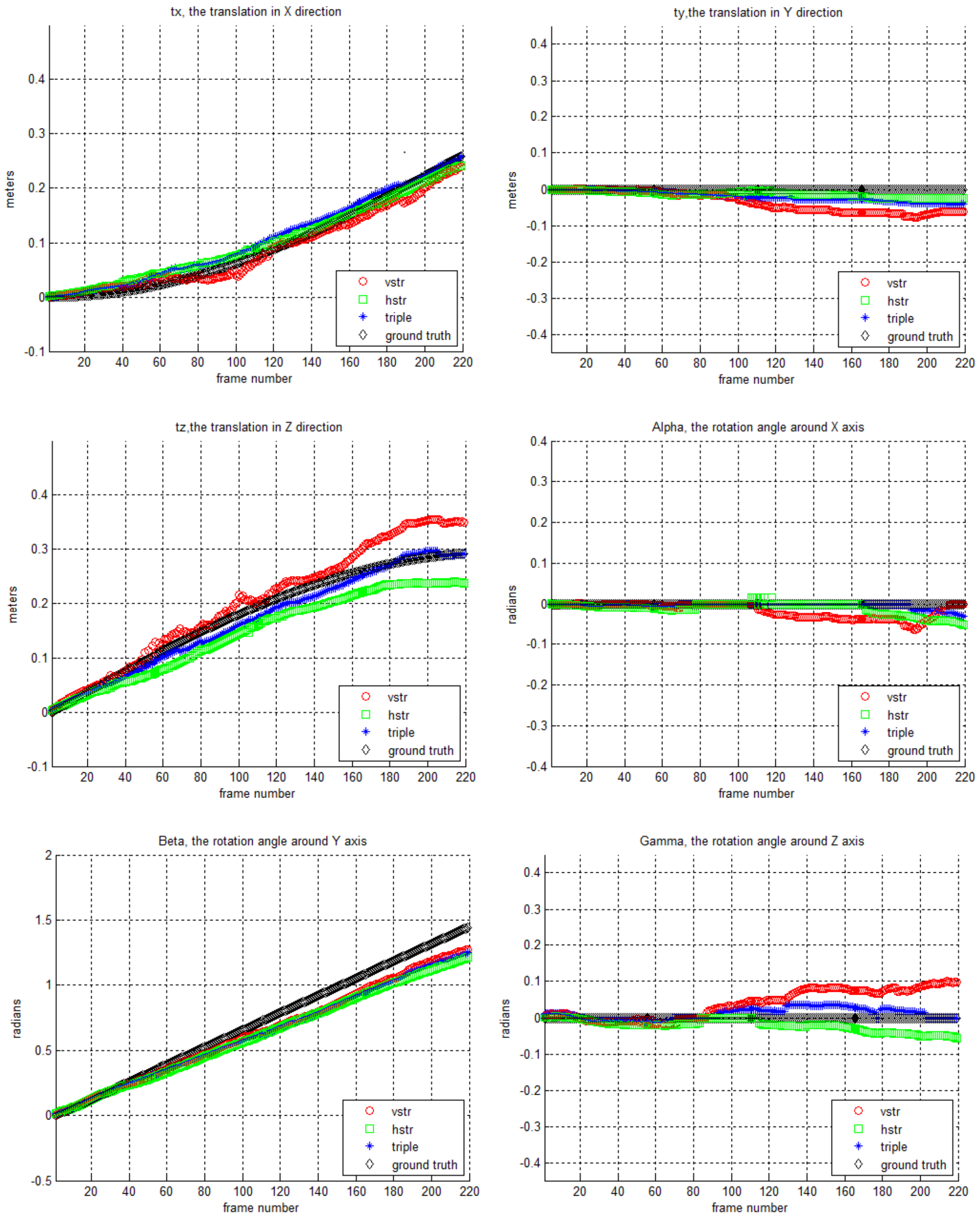

Figure 7. Results of real experiment (curve pattern), ‘v_{str}’ is the vertical stereo, ‘h_{str}’ is the horizontal stereo, and the ‘triple’ is the three cameras. Remarkable differences are noted, more details in section 4.

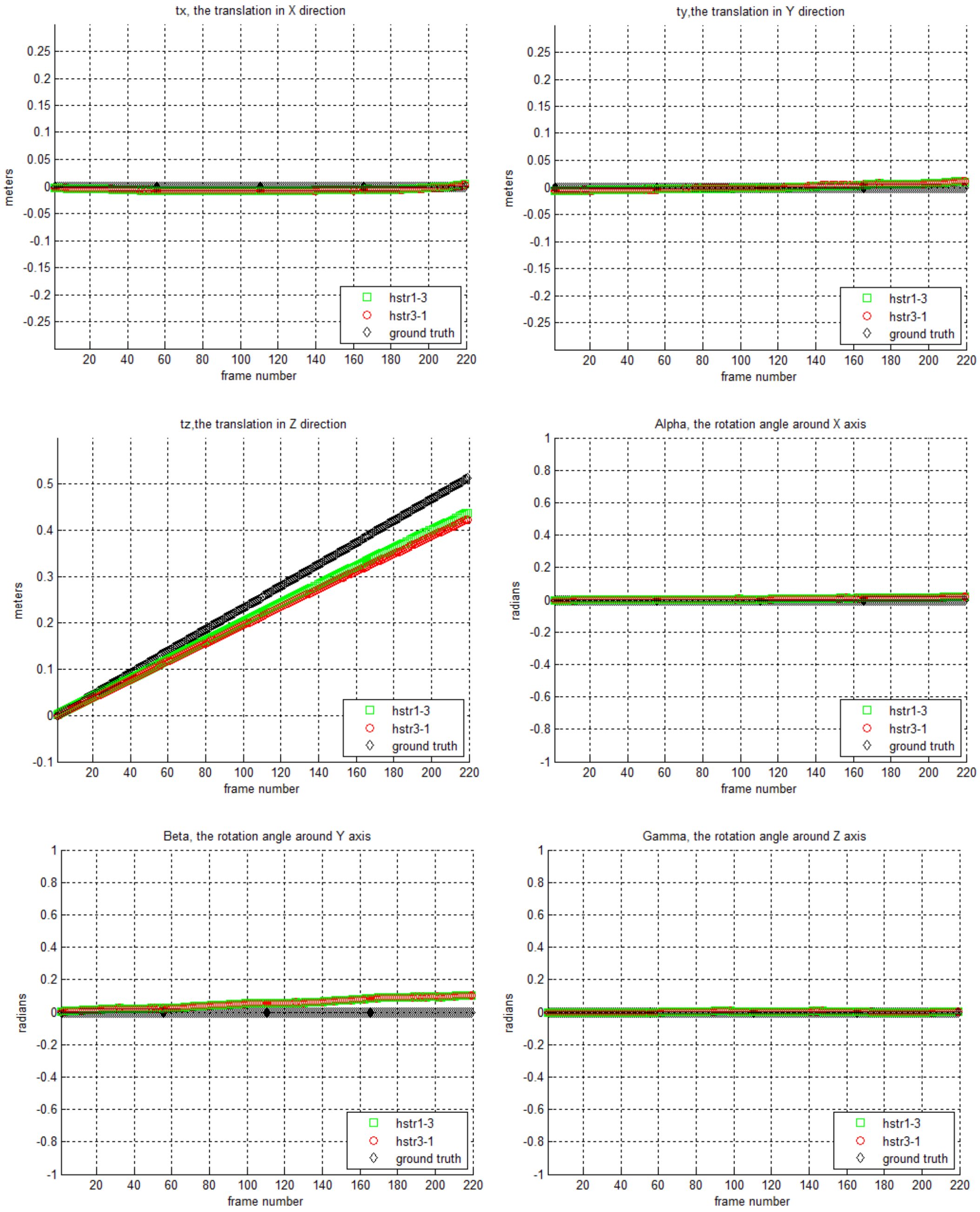

Figure 8. Results of real experiment (pure translation pattern), ‘h_{str1-3}’ is the same as ‘h_{str}’, while ‘h_{str3-1}’ has camera 3 as the reference of the horizontal stereo pair. No difference is noticed for this pattern.

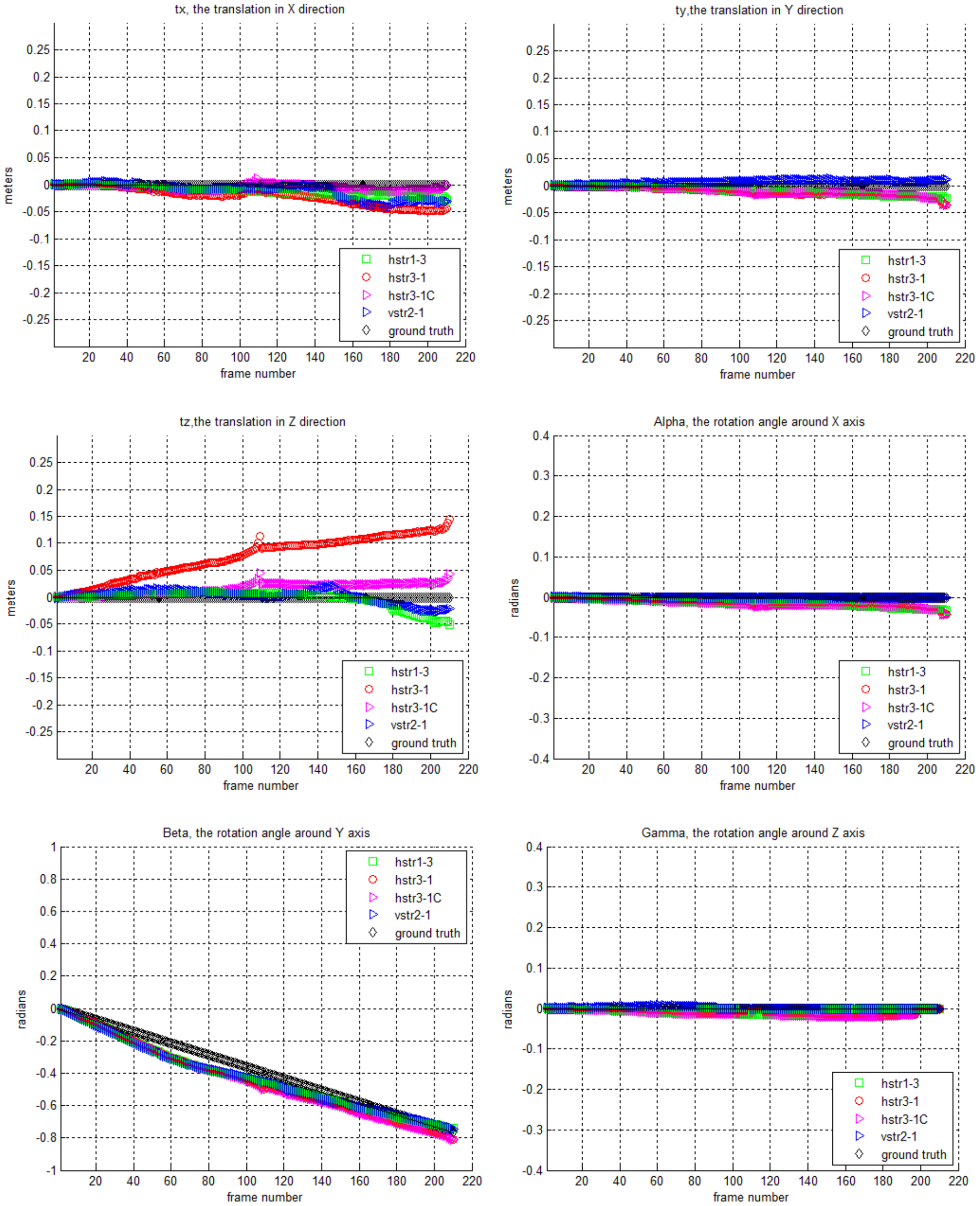

Figure 9. Results of real experiment (pure rotation pattern), ‘h_{str1-3}’ is same as ‘h_{str}’, while ‘h_{str3-1}’ has camera 3 as reference, corrected in ‘$h_{str3-1C}$’, and ‘v_{str2-1}’ is the vertical stereo with camera 2 as reference.

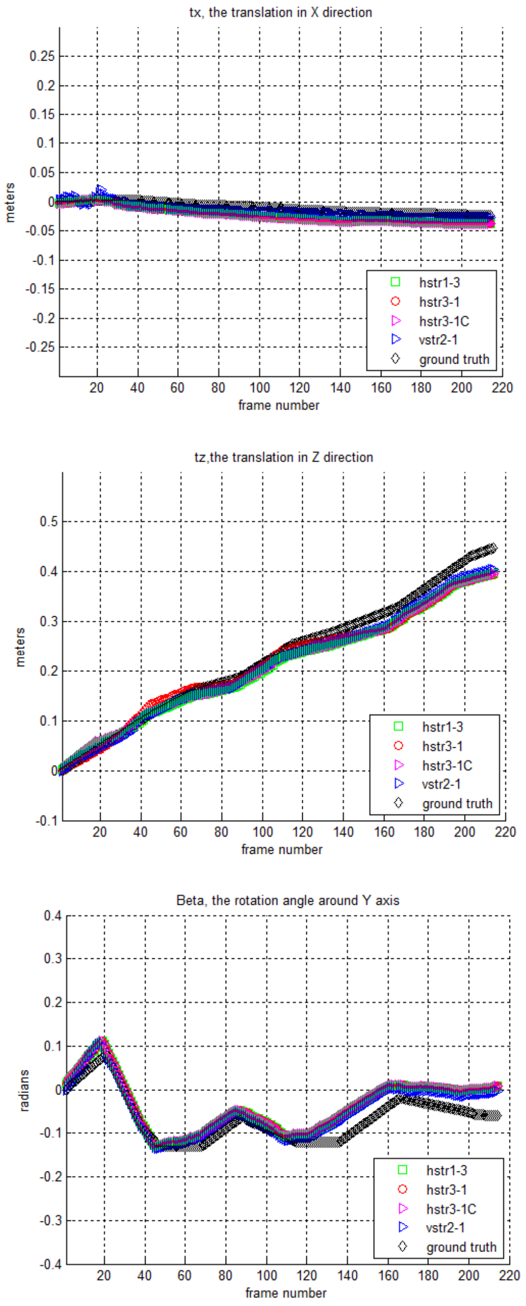

Figure 10. Results of real experiment (mixed pattern), ‘h_{str1-3}’ is same as ‘h_{str}’, while ‘h_{str3-1}’ has camera 3 as reference, corrected in ‘$h_{str3-1C}$’, and ‘v_{str2-1}’ is the vertical stereo with camera 2 as reference.

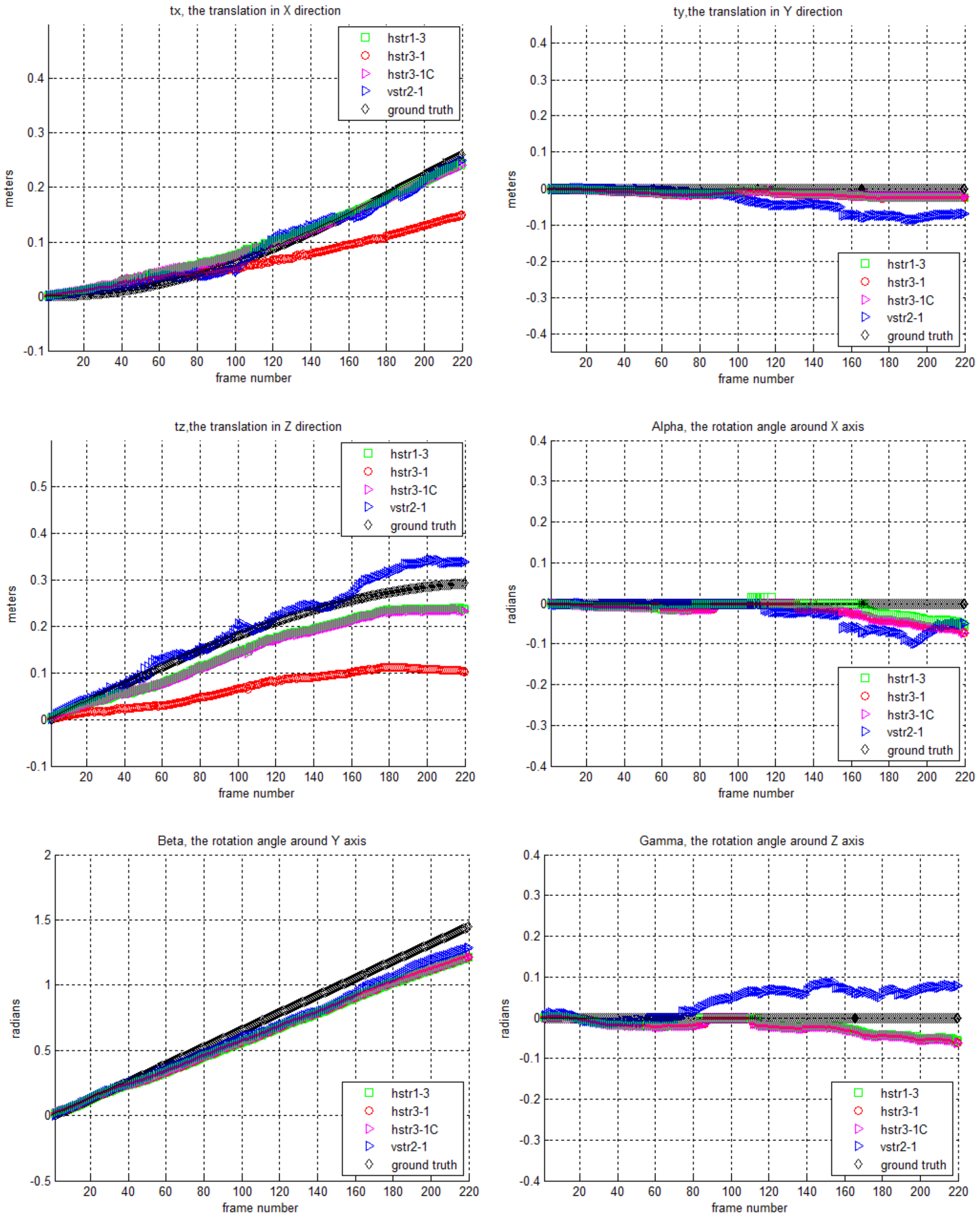

Figure 11. Results of real experiment (curve pattern), ‘h_{str1-3}’ is same as ‘h_{str}’, while ‘h_{str3-1}’ has camera 3 as reference, corrected in ‘$h_{str3-1C}$’, and ‘v_{str2-1}’ is the vertical stereo with camera 2 as reference.