Stable Vision-Aided Navigation for Large-Area Augmented Reality

Taragay Oskiper* Han-Pang Chiu† Zhiwei Zhu‡ Supun Samaresekera§ Rakesh Kumar¶

SRI International Sarnoff

Abstract

In this paper, we present a unified approach for a drift-free and jitter-reduced vision-aided navigation system. This approach is based on an error-state Kalman filter algorithm using both relative (local) measurements obtained from image based motion estimation through visual odometry, and global measurements as a result of landmark matching through a pre-built visual landmark database. To improve the accuracy in pose estimation for augmented reality applications, we capture the 3D local reconstruction uncertainty of each landmark point as a covariance matrix and implicity rely more on closer points in the filter. We conduct a number of experiments aimed at evaluating different aspects of our Kalman filter framework, and show our approach can provide highly-accurate and stable pose both indoors and outdoors over large areas. The results demonstrate both the long term stability and the overall accuracy of our algorithm as intended to provide a solution to the camera tracking problem in augmented reality applications.

Index Terms: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, Augmented, and Virtual Reality; I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Sensor Fusion

1 Introduction

Navigation for augmented reality systems using head mounted displays (HMDs) has very demanding requirements: It must estimate highly accurate 3D location and 3D orientation of the user's head in real time. This is required by the system to know where to insert the synthetic actors and objects in the HMD. In addition, the inserted objects must appear stable. The presence of drift or jitter on inserted objects will disturb the illusion of mixture between rendered and real world for the user.

In this paper, we present our work on the real-time navigation component of an augmented reality training and gaming system, which can be used both indoors and outdoors over large areas. The system uses computer graphics and see-through head mounted displays to insert virtual actors, objects and sound effects into the scene as viewed by each user. The virtual actors respond in realistic ways to actions of the user, taking cover, or firing back, or milling as crowds etc. In this system, there is no need to instrument the training/gaming facility; the individual users wear the primary hardware. The augmented reality interface subsystem worn by each individual user consists of a helmet mounted sensor platform (Figure 1) with front and back facing stereo cameras, inertial measurement unit (IMU), a compact computer mounted on his backpack and a see through HMD.

We introduce a new Kalman filter framework to fuse IMU data, the local measurements from the distributed aperture visual odometry algorithm with front and back facing stereo cameras [17], and the global measurements from the visual landmark matching module [27]. Our Kalman filter eliminates the constant velocity process model used in [17], and is able to incorporate the global measurements which are 3D to 2D feature point correspondences between the pre-built landmark database and given query image. As shown in Figure 2, the final estimated global pose is generated from the filter instead of the landmark matcher as was done in [27].

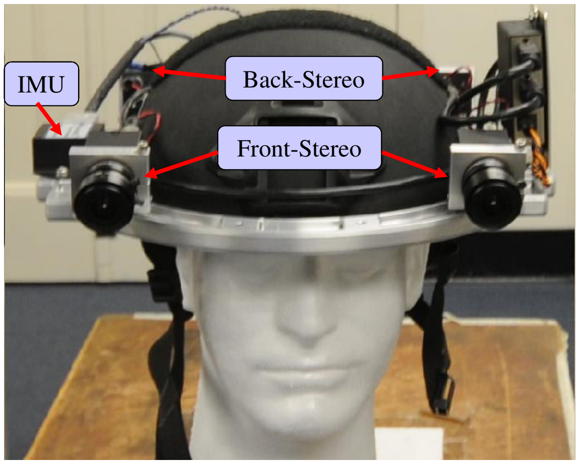

Figure 1: Our helmet mounted sensor platform.

This unified approach to fuse all the measurement data allows for better handling of the uncertainty propagation through the whole system. It is not possible in the framework of [27] in which the Kalman filter output was used to locally propagate the navigation solution from one landmark match instance to another with the pose solution based on landmark matching effectively resetting the filter output. By fusing both inertial and vision measurements, our system is also more robust under challenging conditions where there are insufficient visual clues to rely on.

The other advantage of our work is to eliminate a trade-off problem in [27]: landmark matching between the pre-built database and the current query frame provides global fixes to prevent the estimated poses from drifting during online tracking, but often lacks precision which results in jitter. We found that the accuracy of pose estimation from the landmark matcher decreases if there are fewer landmark point matches closer to the camera where the depth estimation is more accurate. To reduce the jitter, we capture the 3D local reconstruction uncertainty of each landmark point as a covariance matrix and implicity rely more on closer points as global measurements in the Kalman filter. This provides more accurate and stable pose estimation to fulfill the demanding requirements for augmented reality systems.

2 Previous Work

Prototype augmented-reality training/gaming systems require significant infrastructure. For example, a few systems use video projectors to project images of virtual actors on walls of rooms within a facility. Existing systems also have a limited ability to track users,

*e-mail:taragay.oskiper@sri.com
†e-mail:han-pang.chiu@sri.com
‡e-mail:zhiwei.zhu@sri.com
§e-mail:supun.samarasekera@sri.com
¶e-mail:rakesh.kumar@sri.com

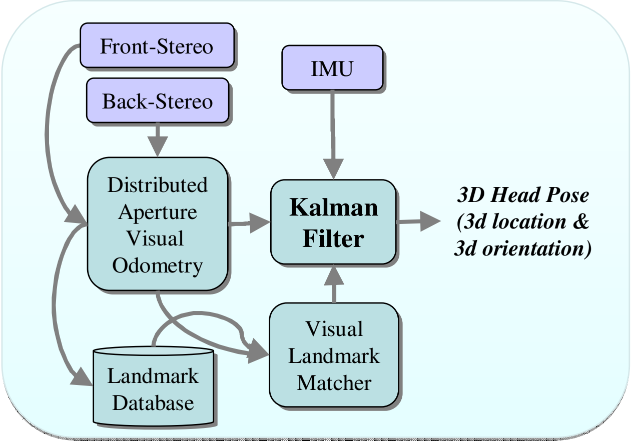
Front-Stereo
Back-Stereo
IMU
Distributed Aperture Visual Odometry
Kalman Filter
3D Head Pose (3d location & 3d orientation)
Visual Landmark Matcher
Landmark Database

Figure 2: System block diagram.

and to adapt virtual actions or their projection to the movements of the users. GPS-based systems [22, 10] may be used for providing location outdoors. However, the performance of these outdoor-only systems decreases in challenging situations. Recent vision-based systems [2, 12, 25, 6, 11] can provide accurate tracking indoors even on mobile phones. However, they still have difficulties to handle occlusion or feature-less environment. Overall, providing high accuracy tracking over large indoor and outdoor areas (multiple square miles) is a very challenging problem.

Real-time tracking by fusing visual and inertial sensors has been studied for many years with numerous applications in robotics, vehicle navigation and augmented reality. However, it is still unclear how to best combine the information from these sensors. Since inertial sensors are suited for handling no vision situations due to fast motion or occlusion, many researchers use inertial data as backup [1] or take only partial information (gyroscopes) from IMU [26, 19, 9] to support vision-based tracking systems.

To better exploit inertial data, several researchers use an extended Kalman filter to fuse all measurements uniformly to a pose estimate. They combine the filter with vision-tracking techniques based on artificial markers [5], feature points, or lines. These systems show that the vision measurements effectively reduce the errors accumulated from IMU. However, most of them conduct experiments on either synthetic data [18] or simulated vision measurements [8]. Some systems provide results on realistic data, but within simple test environments [23] or small rooms [3]. Moreover, they cannot eliminate the problem of long term drift over large areas inherent in inertial-based navigation platform.

Due to recent advances in the image searching techniques, real-time landmark matching with a large landmark database has become possible [16, 24]. Zhu et al. [27] integrated visual landmark matching to a pre-built landmark database in a visual-inertial navigation system. The continuously updating landmark matching corrects the long term drift in the system, and thus improves the overall performance. However, they only used IMU data for transition between views where visual features are lost due to fast motion or bad illumination. Moreover, landmark matching often lacks high-precision in pose estimation, which is required for augmented reality applications.

There are two major differences between our work and other visual-inertial navigation systems. First, we adopt the so called ”error-state” formulation [21] in the extended Kalman filter. Under this representation, there is no need to specify an explicit dynamic motion model which was used in [17] for a given sensor platform. The filter dynamics follow from the IMU error propagation equations which evolve slowly over time and therefore are more amenable to linearization. The measurements to the filter consist of the differences between the inertial navigation solution as obtained by solving the IMU mechanization equations and the external source data, which in our case is the relative pose information provided by visual odometry algorithm and global measurements provided by the visual landmark matching process (Figure 3).

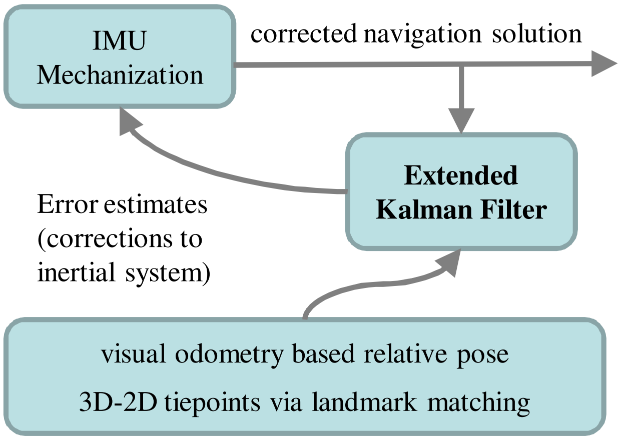
IMU Mechanization
corrected navigation solution
Extended Kalman Filter
Error estimates (corrections to inertial system)
visual odometry based relative pose
3D-2D tiepoints via landmark matching

Figure 3: Error-state Extended Kalman Filter block diagram with local and global external measurements.

Second, our Kalman filter framework incorporates two complementary vision measurements based on state-of-the-art vision tracking techniques. Relative pose measurements based on feature tracking between adjacent frames are usually located very precisely. Therefore, they do not jitter but suffer from drift or loss of track. Landmark matching [27] provides correspondences between fixed 3D features in a pre-built database and 2D points on the query frame. These measurements avoid drift but cause jitter. To make the outputted pose not only accurate but also stable, we fuse both local and global information in the filter.

3 Extended Kalman Filter Process Model

In our extended Kalman filter, we denote the ground (global coordinate frame) to camera pose as $\mathbf{P}_{GC} = [\mathbf{R}_{GC} \;\; \mathbf{T}_{GC}]$ such that a point $\mathbf{X}_G$ expressed in the ground frame can be transferred to the camera coordinates by $\mathbf{X}_C = \mathbf{R}_{GC}\mathbf{X}_G + \mathbf{T}_{GC}$. Accordingly, $\mathbf{T}_{GC}$ represents the ground origin expressed in the camera coordinate frame, whereas $\mathbf{T}_{CG} = -\mathbf{R}_{GC}^T\mathbf{T}_{GC}$ is the camera location in the ground coordinate frame.

In this paper, without loss of generality and to keep the notation simple, we will assume that the camera and IMU coordinate systems coincide so that $\mathbf{P}_{GI} = \mathbf{P}_{GC}$. In reality we use an extrinsic calibration procedure to determine the camera to IMU pose $\mathbf{P}_{CI}$, (front left stereo camera is chosen as the master) as developed in [15] and distinguish between $\mathbf{P}_{GI} = \mathbf{P}_{CI}\mathbf{P}_{GC}$ and $\mathbf{P}_{GC}$.

The total (full) states of the filter consist of the camera location $\mathbf{T}_{CG}$, the gyroscope bias vector $\mathbf{b}_g$, velocity vector $\mathbf{v}$ in global coordinate frame, accelerometer bias vector $\mathbf{b}_a$ and ground to camera orientation $\mathbf{q}_{GC}$, expressed in terms of the quaternion representation for rotation, such that $\mathbf{R}_{GC} = (|q_0|^2 - \|\overrightarrow{\mathbf{q}}\|^2)\mathbf{I}_{3\times3} + 2\overrightarrow{\mathbf{q}}\,\overrightarrow{\mathbf{q}}^T - 2q_0\left[\overrightarrow{\mathbf{q}}\right]_\times$, with $\mathbf{q}_{GC} = [q_0 \;\; \overrightarrow{\mathbf{q}}^T]^T$ and $\left[\overrightarrow{\mathbf{q}}\right]_\times$ denoting the skew-symmetric matrix formed by $\overrightarrow{\mathbf{q}}$. For quaternion algebra, we follow the notation and use the frame rotation perspective as described in [13]. Hence, the total (full) state vector is given by

$$\mathbf{s} = [\mathbf{q}_{GC}^T \quad \mathbf{b}_g^T \quad \mathbf{v}^T \quad \mathbf{b}_a^T \quad \mathbf{T}_{CG}^T]^T \,. \tag{1}$$

We use the corresponding system model for the state time evolution

$$\dot{\mathbf{q}}_{GC}(t) = \frac{1}{2}(\mathbf{q}_{GC}(t) \otimes \omega(t)), \quad \dot{\mathbf{b}}_g(t) = \mathbf{n}_{wg}(t)$$
$$\dot{\mathbf{v}}(t) = \mathbf{a}(t), \quad \dot{\mathbf{b}}_a(t) = \mathbf{n}_{wa}(t), \quad \dot{\mathbf{T}}_{CG}(t) = \mathbf{v}(t)$$

where $\mathbf{n}_{wg}$ and $\mathbf{n}_{wa}$ are modeled as white Gaussian noise, and $\mathbf{a}(t)$ is camera acceleration in global coordinate frame, and $\omega(t)$ is the rotational velocity in camera coordinate frame. Gyroscope and accelerometer measurements of these two vectors are modeled as:

$$\omega_m(t) = \omega(t) + \mathbf{b}_g(t) + \mathbf{n}_g(t) \tag{2}$$
$$\mathbf{a}_m(t) = \mathbf{R}_{GC}(t)(\mathbf{a}(t) - \mathbf{g}) + \mathbf{b}_a(t) + \mathbf{n}_a(t) \tag{3}$$

where $\mathbf{n}_g$ and $\mathbf{n}_a$ are modeled as white Gaussian noise and $\mathbf{g}$ is the gravitational acceleration expressed in the global coordinate frame.

State estimate propagation is obtained by the IMU mechanization equations

$$\dot{\hat{\mathbf{q}}}_{GC}(t) = \frac{1}{2}(\hat{\mathbf{q}}_{GC}(t) \otimes \hat{\omega}(t)) \tag{4}$$
$$\dot{\hat{\mathbf{v}}}(t) = \hat{\mathbf{R}}^T_{GC}(t)\hat{\alpha}(t) + \mathbf{g}, \tag{5}$$
$$\dot{\hat{\mathbf{x}}}(t) = \hat{\mathbf{v}}(t), \qquad \dot{\hat{\mathbf{b}}}_g(t) = 0, \qquad \dot{\hat{\mathbf{b}}}_a(t) = 0 \tag{6}$$

with $\hat{\omega}(t) = \omega_m(t) - \hat{\mathbf{b}}_a(t)$, and $\hat{\alpha}(t) = \mathbf{a}_m(t) - \hat{\mathbf{b}}_a(t)$.

We solve the above system by fourth-order Runge-Kutta numerical integration method. The Kalman filter error state consists of

$$\delta\mathbf{s} = [\delta\Theta^T \quad \delta\mathbf{b}_g^T \quad \delta\mathbf{v}^T \quad \delta\mathbf{b}_a^T \quad \delta\mathbf{T}_{CG}^T]^T \tag{7}$$

according to the following relation between the total state and its inertial estimate

$$\mathbf{q}_{GC} = \hat{\mathbf{q}}_{GC} \otimes \delta\mathbf{q}_{GC}, \text{ with } \delta\mathbf{q}_{GC} \simeq [1 \quad \delta\Theta^T/2]^T$$
$$\mathbf{b}_g(t) = \hat{\mathbf{b}}_g(t) + \delta\mathbf{b}_g(t), \ \mathbf{b}_a(t) = \hat{\mathbf{b}}_a(t) + \delta\mathbf{b}_a(t)$$
$$\mathbf{v}(t) = \hat{\mathbf{v}}(t) + \delta\mathbf{v}(t), \ \mathbf{T}_{CG}(t) = \hat{\mathbf{T}}_{CG}(t) + \delta\mathbf{T}_{CG}(t)$$

based on which we obtain (after some algebra) the following dynamic process model for the error state:

$$\dot{\delta}\mathbf{s} = \mathbf{F}\delta\mathbf{s} + \mathbf{G}\mathbf{n} \tag{8}$$

where

$$\mathbf{F} = \begin{bmatrix} -[\hat{\omega}]_\times & -\mathbf{I}_3 & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \\ -\hat{\mathbf{R}}^T_{GC}[\hat{\alpha}]_\times & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & -\hat{\mathbf{R}}^T_{GC} & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{I}_3 & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \end{bmatrix},$$

$$\mathbf{n} = \begin{bmatrix} \mathbf{n}_g \\ \mathbf{n}_{wg} \\ \mathbf{n}_a \\ \mathbf{n}_{wa} \end{bmatrix}, \text{ and } \mathbf{G} = \begin{bmatrix} -\mathbf{I}_3 & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{I}_3 & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & -\hat{\mathbf{R}}^T_{GC} & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{I}_3 \\ \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \end{bmatrix}$$

4 Visual Odometry and Landmark Matching Measurement Model

The two vision measurements to our Kalman filter are from visual odometry and landmark matching. Our system captures the synchronized stereo image pairs from cameras and IMU measurements in real time. The captured stereo pairs are sent to the visual odometry module that extracts Harris corners [7] from each image and estimates the initial camera pose (local pose) from the extracted Harris corners. Once the local visual odometry poses are estimated, the visual odometry module sends the left images together with its extracted Harris corners and the estimated pose to the landmark matching module.

To incorporate visual odometry poses that are relative in nature, we apply the same stochastic cloning approach developed in [20] for our measurement model. In particular, we denote $\mathbf{P}_{1,2}$ as the visual odometry pose estimate between two time instances 1 and 2, and let the corresponding pose components of the state be denoted by $\mathbf{P}_{G,1}$ and $\mathbf{P}_{G,2}$. Then defining $\mathbf{T}_{2,1} = \mathbf{R}_{G,1}(\mathbf{T}_{2,G} - \mathbf{T}_{1,G})$, and $\mathbf{q}_{1,2} = \mathbf{q}^{-1}_{G,1}\mathbf{q}_{G,2}$, and after lengthy algebra as similar to [20], we obtain the following measurement equations

$$\delta\mathbf{z}_T = \left[\hat{\mathbf{R}}_{G,1}(\hat{\mathbf{T}}_{2,G} - \hat{\mathbf{T}}_{1,G})\right]_\times \delta\Theta_{G,1} + \hat{\mathbf{R}}_{G,1}\delta\mathbf{T}_{2,G} \tag{9}$$
$$- \hat{\mathbf{R}}_{G,1}\delta\mathbf{T}_{1,G} + \nu_T \tag{10}$$

and

$$\delta\mathbf{z}_q = 1/2\,\hat{\mathbf{R}}^T_{1,2}\delta\Theta_{G,2} - 1/2\,\delta\Theta_{G,1} + \nu_q \tag{11}$$

where ν_T and ν_q are the Gaussian noise in translation and rotation associated with the visual odometry pose solution. These measurements are a function of the propagated error-state $\delta\mathbf{s}_2$ and the cloned error-state $\delta\mathbf{s}_1$ from previous time instance, which require modifications to the Kalman filter update equations (cf. [20]).

For landmark matching, after receiving the extracted Harris corners from visual odometry module, the HOG descriptor [4] is extracted for each Harris corner from the received left image. Once the HOG descriptors are extracted, the normalized 2D image coordinates, 3D coordinates, and the HOG descriptors of the corners together with the camera pose is formed as a landmark shot. The landmark shot will serve as a query landmark shot and search the landmark database for a list of potential similar landmark shots and then an image-based 2D matching is performed to refine the potential landmark shot list. Once a matched landmark shot is found, the global camera pose of that shot is also computed.

Thus, for landmark matching, given a query image, landmark matching returns the found landmark shot from the database establishing the 2D to 3D point correspondences between the query image features and the 3D local point cloud, as well as the camera pose $\mathbf{P}_{GL}$ belonging to that shot. To incorporate landmark matching into our filter, every 3D local landmark point $\mathbf{X}$ is transferred to the global coordinate system via

$$\mathbf{Y} = \mathbf{R}_{LG}\mathbf{X} + \mathbf{T}_{LG} \tag{12}$$

which can be written under small error assumption as

$$\hat{\mathbf{Y}} + \delta\mathbf{Y} \simeq (\mathbf{I} - [\rho]_\times)\hat{\mathbf{R}}_{LG}(\hat{\mathbf{X}} + \delta\mathbf{X}) + \hat{\mathbf{T}}_{LG} + \delta\mathbf{T}_{LG}$$

where ρ is a small rotation vector. Neglecting second order terms results in the following linearization

$$\delta\mathbf{Y} \simeq \hat{\mathbf{R}}_{LG}\delta\mathbf{X} + \left[\hat{\mathbf{R}}_{LG}\hat{\mathbf{X}}\right]_\times \rho + \delta\mathbf{T}_{LG} \tag{13}$$

and letting $\tilde{\mathbf{X}} = \hat{\mathbf{R}}_{LG}\hat{\mathbf{X}}$, the local 3D point covariance, Σ_Y, can be represented in the global coordinate frame in terms of the local reconstruction uncertainty, Σ_X and landmark pose uncertainty in rotation and translation, $\Sigma_{\mathbf{R}_{LG}}$ and $\Sigma_{\mathbf{T}_{LG}}$, as

$$\Sigma_Y \simeq \hat{\mathbf{R}}_{LG}\Sigma_X\hat{\mathbf{R}}^T_{LG} + [\tilde{\mathbf{X}}]_\times\Sigma_{\mathbf{R}_{LG}}[\tilde{\mathbf{X}}]^T_\times + \Sigma_{\mathbf{T}_{LG}}$$

After this transformation, the projective camera measurement model is employed such that for each 3D point $\mathbf{Y}$ obtained above and expressed in the current camera coordinate system as $\mathbf{Z} = [Z_1\ Z_2\ Z_3]^T$, the projection onto the normalized image plane is given by

$$\mathbf{z} = f(\mathbf{Z}) + \nu \text{ with } f(\mathbf{Z}) = [Z_1/Z_3 \quad Z_2/Z_3]^T \tag{14}$$

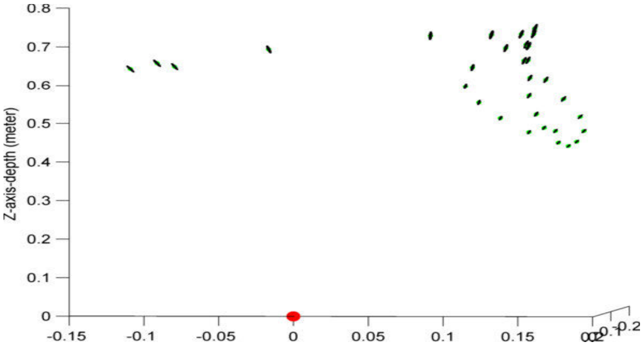

Figure 4: 3D landmark points with associated covariance matrices represented by ellipsoids. The red point (origin) denotes the camera position.

where ν is the feature measurement noise with covariance Σ_ν and

$$\mathbf{Z} = \mathbf{R}_{GC}\mathbf{Y} + \mathbf{T}_{GC} = \mathbf{R}_{GC}(\mathbf{Y} - \mathbf{T}_{CG}) \,. \tag{15}$$

Under small error assumption

$$\hat{\mathbf{Z}} + \delta\mathbf{Z} \simeq (\mathbf{I} - [\delta\Theta]_\times)\hat{\mathbf{R}}_{GC}(\hat{\mathbf{Y}} + \delta\mathbf{Y} - \hat{\mathbf{T}}_{CG} - \delta\mathbf{T}_{CG}) \,.$$

Hence,

$$\delta\mathbf{Z} \simeq \left[\hat{\mathbf{R}}_{GC}(\hat{\mathbf{Y}} - \hat{\mathbf{T}}_{CG})\right]_\times \delta\Theta + \hat{\mathbf{R}}_{GC}(\delta\mathbf{Y} - \delta\mathbf{T}_{CG}) + \nu \,.$$

Accordingly, the measurement equation in the error-states is given by

$$\delta\mathbf{z}_L \simeq \mathbf{H}_L\delta\mathbf{s} + \eta \tag{16}$$

where the measurement Jacobian

$$\mathbf{H}_L = \mathbf{J}_f[\mathbf{J}_\Theta \quad \mathbf{0}_{3\times3} \quad \mathbf{0}_{3\times3} \quad \mathbf{0}_{3\times3} \quad \mathbf{J}_{\delta\mathbf{T}_{CG}}] \tag{17}$$

with

$$\mathbf{J}_f = \begin{bmatrix} 1/\hat{Z}_3 & 0 & -\hat{Z}_1/\hat{Z}_3^2 \\ 0 & 1/\hat{Z}_3 & -\hat{Z}_2/\hat{Z}_3^2 \end{bmatrix} \tag{18}$$

$$\mathbf{J}_\Theta = \left[\hat{\mathbf{R}}_{GC}(\hat{\mathbf{Y}} - \hat{\mathbf{T}}_{CG})\right]_\times, \text{ and } \mathbf{J}_{\delta\mathbf{T}_{CG}} = -\hat{\mathbf{R}}_{GC}$$

and

$$\Sigma_\eta = \mathbf{J}_f[\hat{\mathbf{R}}_{GC}\Sigma_Y\hat{\mathbf{R}}_{GC}^T]\mathbf{J}_f^T + \Sigma_\nu \tag{19}$$

The above is applied to all the point correspondences returned as a result of landmark matching, and all the matrices and vectors are stacked to form the final measurement model equation.

5 3D Local Reconstruction Uncertainty Model

For our augmented reality training/gaming system, the landmark database of the area where the exercise will take place is created before-hand. Mainly, a pan-tilt unit captures both Lidar and stereo imagery while panning full 360 degrees at regularly spaced intervals. All the data is processed offline in a semi-automated manner to produce high fidelity camera poses for each landmark shot stored in the database that includes the image feature coordinates together with their locally triangulated 3D coordinates (expressed in the left camera frame), the 3D local reconstruction uncertainty captured as a covariance matrix, and the associated camera pose for that shot. Also feature descriptors are entered into a vocabulary tree to allow fast indexing during online exercise.

To rely more on closer landmark point measurements in the filter to improve the accuracy in pose estimation, we model the 3D reconstruction uncertainty Σ_X of each 3D local landmark point $\mathbf{P} = [\mathbf{P}_x \ \mathbf{P}_y \ \mathbf{P}_z]^T$ based on methods derived from [14]. The projection from $\mathbf{P}$ to stereo image coordinates is

$$pl = [pl_x \quad pl_y]^T + n = [\mathbf{P}_x/\mathbf{P}_z \quad \mathbf{P}_x/\mathbf{P}_z]^T + n \tag{20}$$

$$pr = [pr_x \quad pr_y]^T + n \tag{21}$$

$$pr_x = \frac{R_1\mathbf{P}_x + R_2\mathbf{P}_y + R_3\mathbf{P}_z + T_1}{R_7\mathbf{P}_x + R_8\mathbf{P}_y + R_9\mathbf{P}_z + T_3}$$

$$pr_y = \frac{R_4\mathbf{P}_x + R_5\mathbf{P}_y + R_6\mathbf{P}_z + T_2}{R_7\mathbf{P}_x + R_8\mathbf{P}_y + R_9\mathbf{P}_z + T_3}$$

$$R = \begin{bmatrix} R_1 & R_2 & R_3 \\ R_4 & R_5 & R_6 \\ R_7 & R_8 & R_9 \end{bmatrix}, \quad T = [T_1 \quad T_2 \quad T_3]^T$$

where pl and pr are 2D image coordinates for left image and right image, R and T denote the stereo extrinsic transformation (rotation matrix and translation vector) from the left camera to the right camera, and n is the zero-mean Gaussian noise.

Given the above measurement model of equations (20) and (21), the estimate of the 3D coordinates of $\mathbf{P}$ can be computed as

$$\begin{aligned} \hat{\mathbf{P}}_x &= pl_x\hat{\mathbf{P}}_z \\ \hat{\mathbf{P}}_y &= pl_y\hat{\mathbf{P}}_z \\ \hat{\mathbf{P}}_z &= \frac{T_1 - T_3 pr_x}{pr_x(R_7 pl_x + R_8 pl_y + R_9) - (R_1 pl_x + R_2 pl_y + R_3)} \end{aligned} \tag{22}$$

Then the covariance matrix of P is

$$\Sigma_X = \mathbf{J}\begin{bmatrix} \mathbf{I}_2 & \mathbf{0}_{2x2} \\ \mathbf{0}_{2x2} & \mathbf{I}_2 \end{bmatrix}\mathbf{J}^T \tag{23}$$

where $\mathbf{J}$ is the matrix of first partial derivatives of (22) which respect to pl_x, pl_y, pr_x, pr_y, or the Jacobian.

Figure 4 shows an example of 3D landmark points with their associated covariance matrices as the reconstruction uncertainty in the real-world coordinate system. We use an 3D ellipsoid around the point's centroid to represent its covariance matrix. The origin of the system is the position of the camera. For points closer to the camera, the ellipsoids are smaller because the 3D reconstructed coordinates are more accurate.

6 Head Prediction for Real-Time Implementation

Even though our unified Kalman filter framework provides very accurate camera poses, in a real-time mixed reality system, one of the most important factors for virtual object insertion is the capability to rapidly compute the camera pose to avoid jitter caused by the time lag between the most recently available camera pose in the system and the rendering time instant.

Figure 5 demonstrates the timing of all the important events in our system. Typically, video frames arrive at a rate much slower than the imu data rate (15Hz versus 100Hz in our system). After each video frame, visual odometry and Kalman filter require a certain processing time (less than frame period) at which point the camera pose corresponding to that frame is available. However, this pose needs to be further corrected by additional processing in order to avoid any problems due to the lag at insertion epochs. For this purpose we use the IMU data that has been buffered in the system between the latest frame time and the current render time. Propagating the most recent pose by integrating all the IMU readings till the current time instant provides the most accurate no-latency camera pose which is sent to the renderer.

This head prediction solution is very effectiveness when the camera poses are lagged within a single frame period (66 milliseconds at 15Hz frame rate). Without this head prediction process, the pose

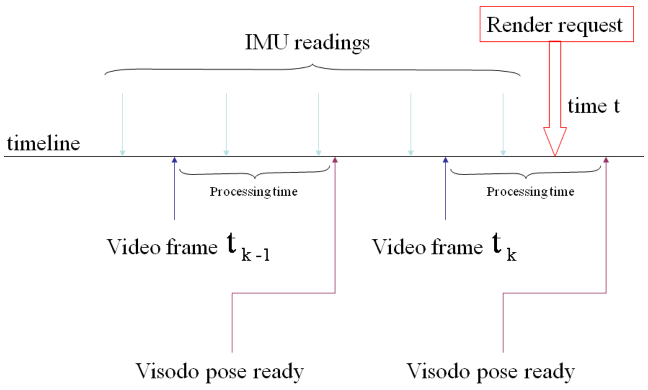

Figure 5: The timing of all the events in our system.

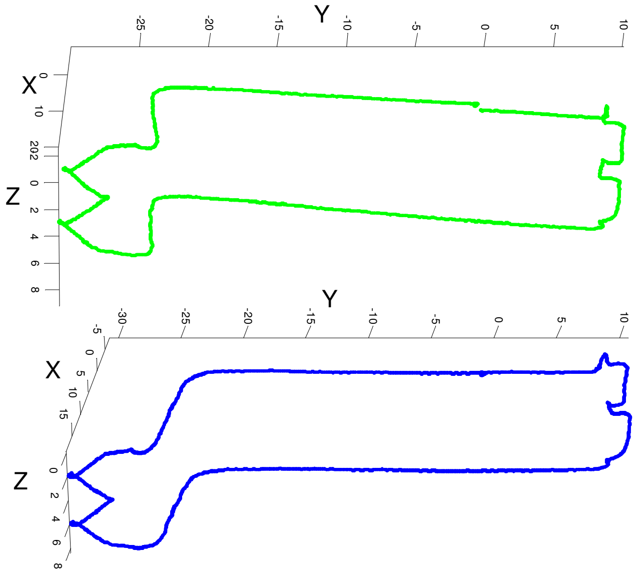

Figure 6: The estimated 3D trajectory by [17] (top) and our Kalman filter (bottom) on the indoor sequence. The total traveled distance is around 157.5 meters.

output will lag the insertion time by a frame period. The virtual object will bounce around significantly rather than stay still in place. However, with our solution, all the jitter is eliminated (see our supplementary videos).

7 Experimental Results

In this section, we report a number of experiments aimed at evaluating different aspects of the performance of our Kalman filter framework. We also demonstrate that our framework can provide highly-accurate real-time tracking both indoors and outdoors over large areas. Compared to [17, 27], we show our navigation system can provide more stable pose estimation to fulfill the demanding requirements for augmented reality applications.

7.1 Performance of fusing visual odometry and inertial data

We first compare the performance between our extended Kalman filter and the method in [17] on the integration of visual odometry and IMU data without landmark matching. For this experiment, we collected two video sequences which demonstrate the exploitation of IMU and visual odometry data.

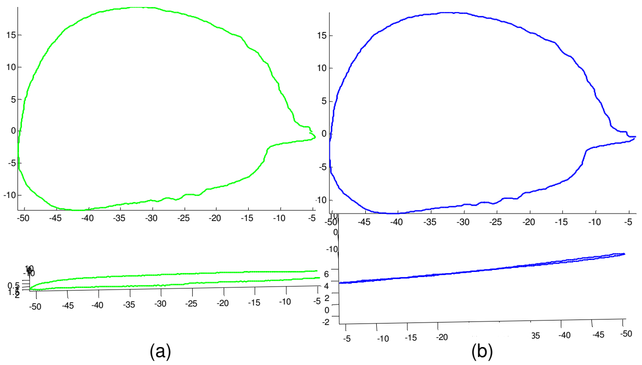

Figure 7: The top and side views of the estimated trajectory by [17] (a) and our Kalman filter (b) on the outdoor sequence. The total traveled distance is around 129 meters.

In the first sequence, the user wearing our hardware system traveled along a predefined closure route inside a building. There are many places in the building where the visual odometry fails to work properly due to poor illuminations or non-textured scenes. For example, there are cases that all the cameras see mostly white walls so that visual odometry cannot estimate pose accurately due to insufficient features.

The user started from a fixed position at the second floor. He walked through the corridor and took the stairs (along Z-axis) to the first floor. Then he went through the corridor on the first floor, took the stair at the other side of the building back to the second floor, and went back to the start position. The whole video sequence is around three and half minutes, and the total traveled distance is around 157.5 meters.

The estimated 3D trajectory by our filter (Figure 6(bottom)) correctly depicts the structure of the two-floor building and the 3D loop closure error is only 0.4639 meters ([17]: 0.6760 meters). Compared to the results by [17], it shows our filter better combines the information from IMU and visual odometry.

For the second sequence, the user walked on the planar ground outdoors and traveled along a closure route. The total traveled distance is around 129 meters. The estimated trajectory by [17] is shown in Figure 7(a). From the side view shown in Figure 7(a), it is clear that there are distance deviations. The 3D loop closure distance is 1.2020 meters. Our Kalman filter (Figure 7(b)), which eliminates the constant velocity process model used in [17], generates a more flat trajectory. The 3D loop closure error by our filter is only 0.3916 meters.

7.2 Performance of fusing local and global measurements

Our Kalman filter allows for better handling of the uncertainty propagation through the whole system, and is able to incorporate the global measurements which are 3D to 2D feature point correspondences from landmark matching [27]. To demonstrate the influence to our filter by incorporating these global measurements, we collected an outdoor sequence while the user wearing our system traveled along a predefined path. We constructed the landmark database of the area where the user would travel before-hand. The user traveled around 256 meters (3 minutes and 40 seconds long) and went back to the starting position. The result (Figure 8) shows that fusing global measurements reduces the 3D loop closure error estimated by our filter from 2.4873 meters to 0.5712 meters.

7.3 Accuracy of fusing local and global measurements

To measure the actual error of the 3D trajectory estimated by our system, we constructed a landmark database inside a rectangle

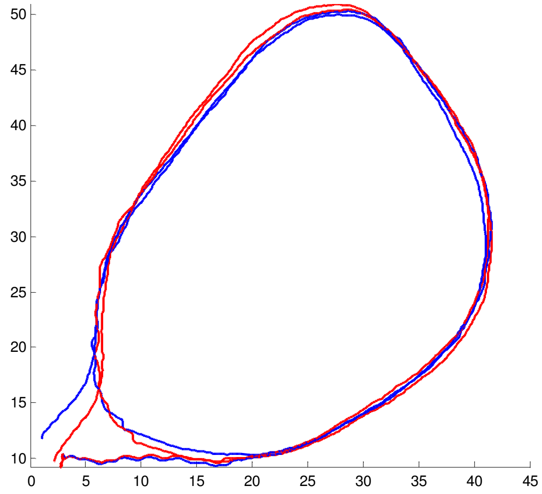

Figure 8: The top view of the estimated trajectories. Blue line shows the estimated 3D trajectory by fusing IMU data and local measurements (relative pose by visual odometry). Red line shows the estimated 3D trajectory by fusing IMU data, local measurements, and global measurements from landmark matching. The total traveled distance is around 256 meters.

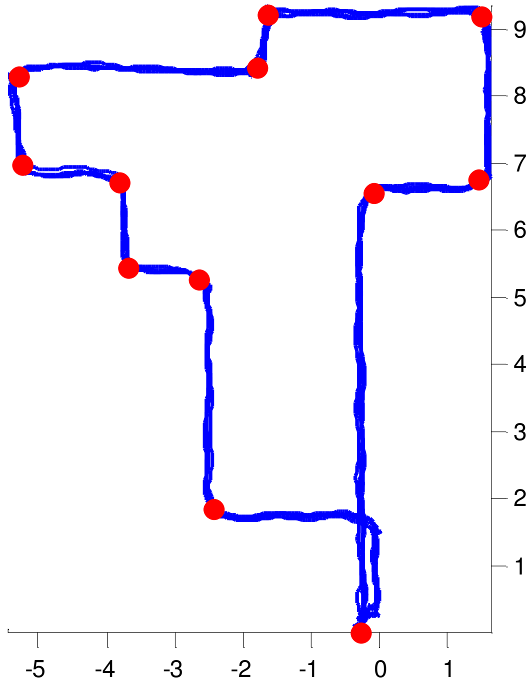

Figure 9: The top view of the estimated trajectory (blue line) on the floor inside a conference room. The 3D trajectory is computed by fusing IMU data, local measurements, and global measurements from landmark matching. The total traveled distance (5 repetitions on the same route) is around 165 meters. We measure the ground truth of 12 positions (red circles) along the trajectory for comparison. The average error on the 12 positions (60 repetitions) is 0.0852 meters.

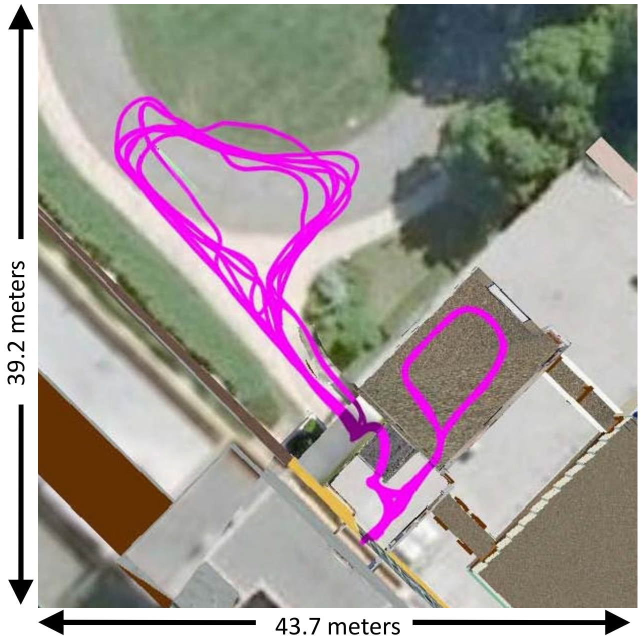

Figure 10: Real-time computed camera trajectory corresponding to a 810 meter long course completed in 16.4 minutes during an online exercise.

room. The width of the room is 7 meters, and the length of the room is 9.5 meters. We marked 12 positions on the floor and retrieved the ground truth of these positions from the database. The user wearing our system then traveled along a predefined path (around 33 meters) which goes through all these 12 positions. We asked the user to walk through the same route five times, and measured the error by comparing our estimated results to the ground truth on these 12 positions. The average error among 60 repetitions is 0.0852 meters.

7.4 Real-time tracking over large areas

To demonstrate that our system can be used both indoors and outdoors over large areas, Figure 10 shows the automatically generated real-time camera trajectory corresponding to an 810 meter course within our campus completed by a user wearing our helmet, backpack system, and a video-see-through HMD. This user walked indoors and outdoors in several loops. The entire area shown in the map is within the pre-built landmark database capture range. The landmark database is loaded in the beginning before the exercise takes place and landmark matches occur whenever a query image is within close proximity to a stored landmark shot in the database.

Figure 11 shows several screen shots corresponding to locations towards the beginning, middle and end of this exercise obtained from our visualization tool which we use to verify the accuracy of the camera pose outputs. This visualization tool uses the camera poses output by our system to render views from a 3D graphical model built upon the same visual data as the landmark database which also forms the global coordinate system. We compare the render views to the actual video images. It is observed that these views are in very good agreement which indicate how precisely the camera is tracked throughout the entire duration of the course.

7.5 Pose estimation for augmented reality

During the same 810 meter course, we inserted virtual actors at particular locations based on the estimated pose and recorded the insertion video which was seen by the user from the video-see-through

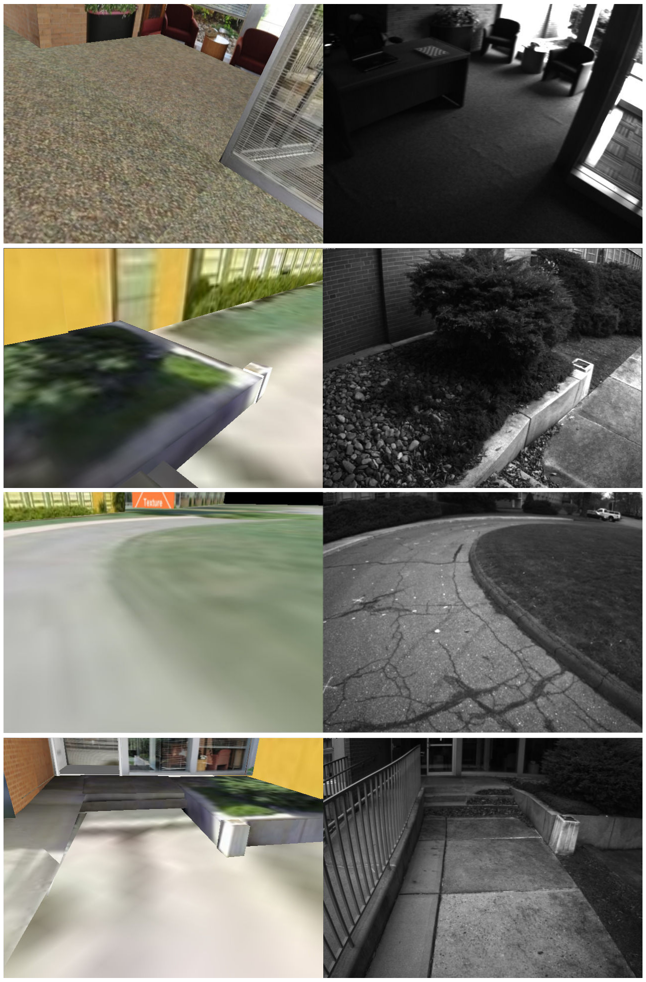

Figure 11: The views rendered from the model using the real-time camera pose estimates by our system for various locations throughout the exercise, together with the real scene views captured by the camera.

HMD. For example, we inserted one virtual actor right outside the stone steps of our building. This video can verify whether the virtual actors are correctly aligned to the real scene based on real-time tracking. The pose estimation from our system needs to be very accurate and stable during the whole course, otherwise it will brake the illusion of mixture between rendered and real world for the user.

Figure 12 shows 8 snapshots of the video when the user went through the entrance of our building at different loops during the 16.4-minute course. The positions of the inserted actor are very consistent in these 8 snapshots. This result demonstrates that our system is able to provide drift-free pose estimation for a long period.

Compared to [27], the major improvement using our unified approach is to provide more stable pose estimation. Figure 13 shows the frame-to-frame pose translation estimated by [27] and our filter respectively. To save the space, we only show the translation over a 450-frame period taken from the whole 16.4 minutes video. Since the walking speed of the user doesn't change much in a very short period (such as one frame, 0.0677 seconds), the translation between frames should be very smooth.

However, in [27], landmark matching disturbs the consistency of pose estimation due to the lack of high-precision. The peaks of the green curve in Figure 13 correspond to the jitter of inserted virtual actors viewed by the user. By capturing the 3D reconstruction uncertainty of landmark points and thus relying more on closer points as global measurements in the Kalman filter, our navigation system can reduce the jitter in pose estimation for augmented reality applications (see our supplementary videos).

8 Conclusion

We presented a unified Kalman filter framework using local and global sensor data fusion for vision aided navigation related to augmented reality applications. We showed results to demonstrate the accuracy and robustness of our system both indoors and outdoors over long duration and distance. Using a pre-built landmark database of the entire exercise area provides precise tracking and eliminates the problem of long term drift inherent in any inertial based navigation platform. Capturing the 3D reconstruction uncertainty of landmark points improves the stability of pose estimation, which is an essential requirement for an augmented reality system.

Future work includes two extensions. First, the current helmet mounted sensor platform worn by the user consists of four cameras (front and back facing stereo cameras), and one inertial measurement unit (IMU). The total weight is around 2 pounds. We plan to lower the weight of the platform by reducing the number of cameras while maintaining the tracking accuracy. We also need to verify the robustness of our method if we use fewer cameras.

Second, the accuracy and frequency of landmark matching when we run the system is influenced by the scene change after we built the database. For example, the furniture arrangement inside a room may change. The natural scene outdoors also may not be the same in different seasons. We want our system can be effective no matter how the scene changes after we built the initial landmark database. One way we are considering is to develop an automated process to update the database periodically, such as using an automated robot planner to move through the site of interest. With these extensions, we believe our system can serve as the basis for a robust, accurate tracking solution for augmented reality gaming and training applications.

References

[1] M. Aron, G. Simon, and B. M. O. Use of inertial sensors to support video tracking. *Computer Animation and Virtual Worlds*, 18, 2007.

[2] C. Arth, D. Wagner, M. Klopschitz, A. Irschara, and D. Schmalstieg. Wide area localization on mobile phones. In *International symposium on mixed and augmented reality*, 2009.

[3] G. Bleser and D. Stricker. Advanced tracking through efficient image processing and visual-inertial sensor fusion. *Computers and Graphics*, 33, 2009.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[5] E. Foxlin and L. Naimark. Vis-tracker:a wearable vision-inertial self-tracker. In *IEEE Virtual Reality*, 2003.

[6] L. Gruber, S. Zollmann, D. Wagner, and D. Schmalstieg. Evaluating the trackability of natural feature-point sets. In *International symposium on mixed and augmented reality*, 2009.

[7] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, 1988.

[8] J. Hol, T. Schon, F. Gustafsson, and P. Slycke. Sensor fusion for augmented reality. In *International conference on information fusion*, 2006.

[9] B. Jiang, U. Neumann, and S. You. A robust hybrid tracking system for outdoor augmented reality. In *IEEE Virtual Reality*, 2004.

[10] J. Karlekar, S. Zhou, W. Lu, Z. Loh, and Y. Nakayama. Positioning, tracking and mapping for outdoor augmentation. In *International symposium on mixed and augmented reality*, 2010.

[11] K. Kim, V. Lepetit, and W. Woo. Keyframe-based modeling and tracking of multiple 3d objects. In *International symposium on mixed and augmented reality*, 2010.

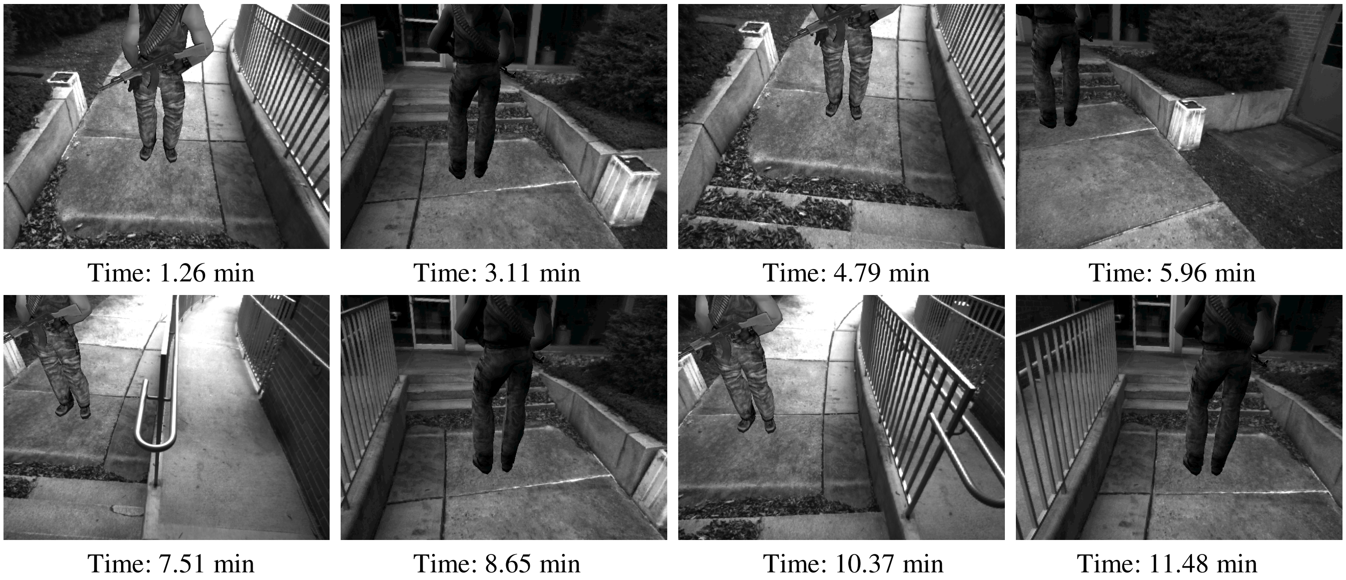
Time: 1.26 min
Time: 3.11 min
Time: 4.79 min
Time: 5.96 min
Time: 7.51 min
Time: 8.65 min
Time: 10.37 min
Time: 11.48 min

Figure 12: 8 snapshots taken from the video when the user went through the entrance of our building at different loops.

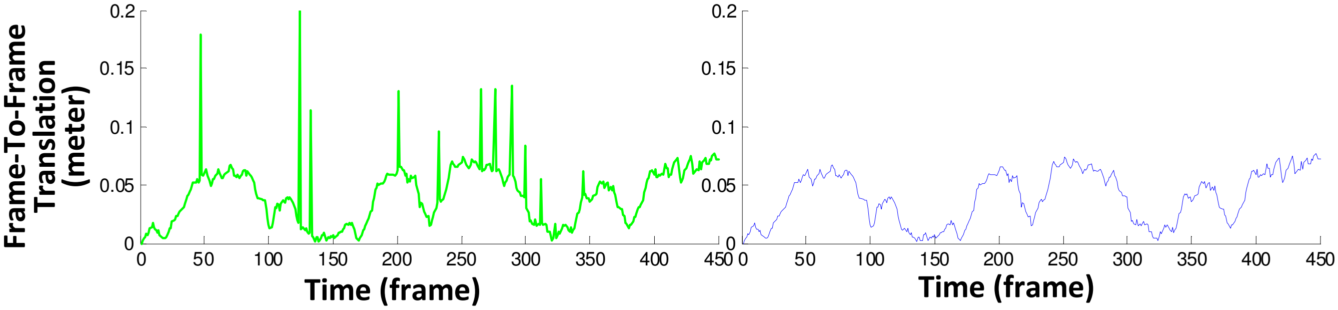

Figure 13: The frame-to-frame estimated translation computed by [27] (green) and our system (blue).

[12] G. Klein and D. Murray. Parallel tracking and mapping on a camera phone. In *International symposium on mixed and augmented reality*, 2009.

[13] J. B. Kuipers. *Quaternions and Rotation Sequences*. Princeton University Press, 1998.

[14] L. Matthies and S. Shafer. Error modeling in stereo navigation. *IEEE Journal of Robotics and Automation*, 3, 1987.

[15] F. M. Mirzaei and S. I. Roumeliotis. A kalman filter-based algorithm for imu-camera calibration: Observability analysis and performance evaluation. *IEEE Transactions on Robotics*, 24(5), 2008.

[16] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *IEEE conference on Computer Vision and Pattern Recognition*, 2006.

[17] T. Oskiper, Z. Zhu, S. Samarasekera, and R. Kumar. Visual odometry system using multiple stereo cameras and inertial measurement unit. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[18] H. Rehbinder and B. Gosh. Pose estimation using line-based dynamic vision and inertial sensors. *IEEE Transactions on Automatic Control*, 48, 2003.

[19] G. Reitmayr and T. Drummond. Going out: robust model-based tracking for outdoor augmented reality. In *International symposium on mixed and augmented reality*, 2006.

[20] S. I. Roumeliotis, A. E. Johnson, and J. F. Montgomery. Augmenting inertial navigation with image-based motion estimation. In *IEEE International Conference on Robotics and Automation*, 2002.

[21] S. I. Roumeliotis, G. S. Sukhatme, and G. Bekey. Circumventing dynamic modeling: Evaluation of the error-state kalman filter applied to mobile robot localization. In *IEEE International Conference on Robotics and Automation*, 1999.

[22] G. Schall, D. Wagner, G. Reitmayr, E. Taichmann, M. Wieser, D. Schmalstieg, and B. Hoftmann-Wellenhof. Global pose estimation using multi-sensor fusion for outdoor augmented reality. In *International symposium on mixed and augmented reality*, 2009.

[23] T. Schon, R. Karlsson, D. Tornqvist, and F. Gustafsson. A framework for simultaneous localization and mapping utilizing model structure. In *International conference on information fusion*, 2007.

[24] S. Se, D. Lowe, and J. Jittle. Vision-based global localization and mapping for mobile robots. *IEEE Transactions on Robotics*, 21(3), 2006.

[25] D. Wagner, D. Schmalstieg, and H. Bischof. Multiple target detection and tracking with guaranteed framerates on mobile phones. In *International symposium on mixed and augmented reality*, 2009.

[26] S. You and U. Neumann. Fusion of vision and gyro tracking for robust augmented reality registration. In *IEEE Virtual Reality*, 2001.

[27] Z. Zhu, T. Oskiper, S. Samarasekera, R. Kumar, and H. S. Sawhney. Real-time global localization with a pre-built visual landmark database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.