Stable Vision-Aided Navigation for Large-Area Augmented Reality

Taragay Oskiper†  Han-Pang Chiu‡  Zhiwei Zhu‡  Supun Samarasekera§  Rakesh Kumar¶
Sarnoff Corporation

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 implicitly 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 [10], and the global measurements from the visual landmark matching module [18]. Our Kalman filter eliminates the constant velocity process model used in [10], 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 [18].

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 [18] in which the Kalman filter output was used to 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 [18]: 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 implicitly 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,
and to adapt virtual actions or their projection to the movements of the
users. GPS-based systems may be used for providing location
outdoors. However, the performance of these outdoor-only systems
decreases in challenging situations. Overall, providing high ac-
curacy 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, ve-
hicle 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
to fast motion or occlusion, many researchers use inertial data as
inertial sensors are suited for handling no vision situations due
to fast motion or occlusion, many researchers use inertial data as


defined by solving the IMU mechanization equations and the exter-
nal 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).

Second, our Kalman filter framework incorporates two comple-
mentary vision measurements based on state-of-the-art vision track-
ing techniques. Relative pose measurements based on feature track-
ing between adjacent frames are usually located very precisely.
Therefore, they do not jitter but suffer from drift or loss of track.
Landmark matching [18] 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.

\section{Extended Kalman Filter Process Model}

In our extended Kalman filter, we denote the ground (global coordi-
nate frame) to camera pose as $P_{GC} = [R_{GC} \quad T_{GC}]$ such that a point
$X_C$ expressed in the ground frame can be transferred to the cam-
ena coordinates by $X_C = R_{GC}X_G + T_{GC}$. Accordingly, $T_{GC}$ re-
present the ground origin expressed in the camera coordinate frame,
whereas $T_{CG} = -R_{GC}^T T_{GC}$ is the camera location in the ground co-
ordinate frame.

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

The total (full) states of the filter consist of the camera location $T_{CG}$, the gyroscope bias vector $b_g$, velocity vector $v$ in global coordi-
inate frame, accelerometer bias vector $b_a$ and ground to camera
orientation $q_{GC}$, expressed in terms of the quaternion representa-
tion for rotation, such that $R_{CG} = (q_{GC}^T - [q_{GC}]^2I_{1 \times 3} + 2q_{GC}^Tq_{GC} - 2q_{GC}^Tq_{GC}q_{GC})$, with $q_{GC} = [q_0, \Vert q \Vert^T]$ and $[q]$ denoting the skew-
symmetric matrix formed by $q$. For quaternion algebra, we follow
the notation and use the frame rotation perspective as described in
[6]. Hence, the total (full) state vector is given by

\begin{equation}
\mathbf{s} = [q_{GC}^T \quad b_g^T \quad v^T \quad b_a^T \quad T_{CG}^T]^T. \tag{1}
\end{equation}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure2}
\caption{System block diagram.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure3}
\caption{Error-state Extended Kalman Filter block diagram with local
and global external measurements.}
\end{figure}
We use the corresponding system model for the state time evolution
\[
\begin{align*}
\dot{q}_{GC}(t) & = \frac{1}{2} (q_{GC}(t) \otimes \dot{\omega}(t)), \\
\dot{\omega}(t) & = n_{\omega}(t), \\
\dot{v}(t) & = a(t), \\
\dot{b}_a(t) & = n_{ba}(t), \\
\dot{T}_{CG}(t) & = v(t)
\end{align*}
\]
where \( n_{\omega} \) and \( n_{ba} \) are modeled as white Gaussian noise, and \( a(t) \) is camera acceleration in camera coordinate frame, and \( \omega(t) \) is the rotational velocity in camera coordinate frame. Gyroscopic and accelerometer measurements of these two vectors are modeled as:
\[
\begin{align*}
\omega_{m}(t) & = \omega(t) + b_{\omega}(t) + n_{\omega}(t) \\
a_{m}(t) & = R_{GC}(t)(a(t) - g) + b_{a}(t) + n_{ba}(t)
\end{align*}
\]
where \( n_{\omega} \) and \( n_{ba} \) are modeled as white Gaussian noise and \( g \) is the gravitational acceleration expressed in the global coordinate frame.

State estimate propagation is obtained by the IMU mechanization equations
\[
\begin{align*}
\dot{q}_{GC}(t) & = \frac{1}{2} (q_{GC}(t) \otimes \dot{\omega}(t)) \\
\dot{\omega}(t) & = R_{GC}^T(t) \dot{\omega}(t) + g, \\
\dot{b}_q(t) & = \dot{v}(t), \\
\dot{b}_a(t) = 0, \\
\dot{b}_\omega(t) = 0
\end{align*}
\]
with \( \dot{\omega}(t) = \omega_{m}(t) - \dot{b}_\omega(t), \) and \( \dot{\omega}(t) = a_{m}(t) - \dot{b}_a(t) \).

We solve the above system by fourth-order Runge-Kutta numerical integration method. The Kalman filter error state consists of
\[
\begin{align*}
\delta s = [\delta \Theta^T \delta b_q^T \delta v^T \delta b_a^T \delta T_{CG}^T]^T
\end{align*}
\]
according to the following relation between the total state and its inertial estimate
\[
\begin{align*}
q_{GC} & = \hat{q}_{GC} \otimes \delta q_{GC}, \\
\dot{b}_q(t) & = \dot{b}_q(t) + \delta b_q(t), \\
\dot{b}_a(t) & = \dot{b}_a(t) + \delta b_a(t), \\
\dot{v}(t) & = \dot{v}(t) + \delta v(t), \\
T_{CG}(t) & = \dot{T}_{CG}(t) + \delta T_{CG}(t)
\end{align*}
\]
Based on which we obtain (after some algebra) the following dynamic process model for the error state:
\[
\delta s = F \delta s + G n
\]
where
\[
F = \begin{bmatrix}
- [\dot{\omega}]_x & -I_3 & 0_{3x3} & 0_{3x3} & 0_{3x3} \\
0_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} \\
-\hat{R}_{GC}^T [\dot{\omega}]_x & 0_{3x3} & 0_{3x3} & -\hat{R}_{GC}^T & 0_{3x3} \\
0_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} & I_3 \\
\hat{n}_f & n_{bq} & n_{va} & n_{wa} & 1
\end{bmatrix}
\]
and
\[
G = \begin{bmatrix}
-1 & 0_{3x3} & 0_{3x3} & 0_{3x3} \\
0_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} \\
0_{3x3} & 0_{3x3} & -\hat{R}_{GC}^T & 0_{3x3} \\
0_{3x3} & 0_{3x3} & 0_{3x3} & I_3 \\
0_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3}
\end{bmatrix}
\]

### 4 Visual Odometry and Landmark Matching Measurement Model

To incorporate visual odometry poses that are relative in nature, we apply the same stochastic cloning approach developed in [13] for our measurement model. In particular, we denote \( P_{1,2} \) 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 \( P_{G,1} \) and \( P_{G,2} \). Then defining \( \hat{q}_{1,2} = P_{G,1}^{-1} q_{G,2} \), and after lengthy algebra as similar to [13], we obtain the following measurement equations
\[
\begin{align*}
\delta z_T & = [\overline{R}_{G,1}(\hat{T}_{2,2} - \hat{T}_{1,1})] \times \delta \Theta_{G,1} + \overline{R}_{G,1} \delta T_{2,2} \quad (9) \\
- \overline{R}_{G,1} \delta T_{1,1} + v_T
\end{align*}
\]
and
\[
\delta z_q = 1/2 \overline{R}_{1,2} \delta \Theta_{G,2} - 1/2 \delta \Theta_{G,1} + v_q \quad (11)
\]
where \( v_T \) and \( v_q \) 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 \( \delta s \) and the cloned error-state \( \delta s_0 \) from previous time instance, which require modifications to the Kalman filter update equations (cf. [13]).

As 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 \( P_{Gi} \) belonging to that shot. First, every 3D local landmark point \( X \) is transferred to the global coordinate system via
\[
Y = R_{LG} X + T_{LG} \quad (12)
\]
which can be written under small error assumption as
\[
\hat{Y} + \delta Y \simeq (I - [\rho]) R_{LG} (X + \delta X) + \delta T_{LG} \quad (13)
\]
and letting \( X = R_{LG} \hat{X} \), the local 3D point covariance, \( \Sigma_{Y} \), can be represented in the global coordinate frame in terms of the local reconstruction uncertainty, \( \Sigma_{x} \) and landmark pose uncertainty in rotation and translation, \( \Sigma_{R_{LG}} \) and \( \Sigma_{T_{LG}} \), as
\[
\Sigma_{Y} \simeq R_{LG} \Sigma_{x} R_{LG}^T + [\hat{X}, \Sigma_{R_{LG}}] (\hat{X}^T + \Sigma_{T_{LG}}) \quad (14)
\]
After this transformation, the projective camera measurement model employed such that for each 3D point \( Y \) obtained above and expressed in the current camera coordinate system as \( Z = [Z_1, Z_2, Z_3]^T \), the projection onto the normalized image plane is given by
\[
z = (Z - [Z_1, Z_2, Z_3])^T \
\]
where \( \nu \) is the feature measurement noise with covariance \( \Sigma_{\nu} \) and
\[
Z = R_{CG} Y + T_{CG} = R_{CG} (Y - T_{CG}) \quad (15)
\]
Under small error assumption
\[
\hat{Z} + \delta Z \simeq (I - [\delta \Theta]) R_{CG} (\hat{Y} + \delta Y - T_{CG} + \delta T_{CG}) \quad (16)
\]
Hence,
\[
\delta Z \simeq [R_{CG} (\hat{Y} - T_{CG})] \times \delta \Theta + \hat{R}_{CG} \delta \Theta \simeq R_{CG} \delta Y + v \quad (17)
\]
Accordingly, the measurement equation in the error-states is given by
\[
\delta z_{FL} \simeq H_{F} \delta s + \nu \quad (18)
\]
where the measurement Jacobian
\[
H_{F} = J_{\delta \theta} \delta \Theta + J_{\sigma_{T_{CG}}} \sigma_{T_{CG}} \quad (19)
\]
with
\[
J_{\delta \theta} = \frac{1}{2 \hat{Z}_3} - \frac{\hat{Z}_1}{2 \hat{Z}_3} \quad (20)
\]
and
\[
\delta \hat{Z}_3 = J_{\sigma_{T_{CG}}} \sigma_{T_{CG}} \quad (21)
\]
where the measurement returns are used 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 beforehand. 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 associated 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 $\Sigma_X$ of each 3D local landmark point $P = [p_x, p_y, p_z]^T$ based on methods derived from [7]. The projection from $P$ to stereo image coordinates is

$$pl = [pl_x\; pl_y]^T + n = [P_x/P_z\; P_y/P_z]^T + n \tag{20}$$

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

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 $P$ can be computed as

$$\hat{P}_x = pl_x \hat{p}_z$$
$$\hat{P}_y = pl_y \hat{p}_z$$
$$\hat{P}_z = \frac{T_1 - T_3 pr_x}{pr_x(R_1 pl_x + R_3 pl_y + R_9) - (R_1 pl_x + R_2 pl_y + R_5)} \tag{22}$$

Then the covariance matrix of $P$ is

$$\Sigma_X = J \begin{bmatrix} I_2 & 0_{2\times 2} & I_2 \end{bmatrix} J^T \tag{23}$$

where $J$ 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 a 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.

If the camera poses are lagged by a single frame period (66 milliseconds at 15Hz frame rate), one can notice that without the head prediction process the virtual object bounces around significantly rather than stay still in place. However, with head prediction, all the jitter arising from pose latency is eliminated (see 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 [10, 18], 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 [10] 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.

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 ([10]: 0.6760 meters). Compared to the results by [10], 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 our filter (Figure 7(a)) 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 [10], generates a more flat trajectory. The 3D loop closure error by our filter is only 0.3916 meters.

Figure 6: The estimated 3D trajectory by [10] (top) and our Kalman filter (bottom) on the indoor sequence. The total traveled distance is around 157.5 meters.

Figure 7: The top and side views of the estimated trajectory by [10] (a) and our Kalman filter (b) on the outdoor sequence. The total traveled distance is around 129 meters.

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.
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 [18]. 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 beforehand. 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 Real-time tracking over large areas

To demonstrate that our system can be used both indoors and outdoors over large areas, Figure 9 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 10 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.

7.4 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 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 11 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 [18], the major improvement using our unified approach is to provide more stable pose estimation. Figure 12 shows the frame-to-frame pose translation estimated by [18] 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 [18], landmark matching disturbs the consistency of pose estimation due to the lack of high-precision. The peaks of the green curve in Figure 12 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.

REFERENCES


