Augmented Reality Binoculars

Taragay Oskiper, Mikhail Sizintsev, *Member, IEEE*, Vlad Branzoi, *Member, IEEE*, Supun Samarasekera, *Member, IEEE*, and Rakesh Kumar, *Member, IEEE*

Abstract—In this paper we present an augmented reality binocular system to allow long range high precision augmentation of live telescopic imagery with aerial and terrain based synthetic objects, vehicles, people and effects. The inserted objects must appear stable in the display and must not jitter and drift as the user pans around and examines the scene with the binoculars. The design of the system is based on using two different cameras with wide field of view and narrow field of view lenses enclosed in a binocular shaped shell. Using the wide field of view gives us context and enables us to recover the 3D location and orientation of the binoculars much more robustly, whereas the narrow field of view is used for the actual augmentation as well as to increase precision in tracking. We present our navigation algorithm that uses the two cameras in combination with an inertial measurement unit and global positioning system in an extended Kalman filter and provides jitter free, robust and real-time pose estimation for precise augmentation. We have demonstrated successful use of our system as part of information sharing example as well as a live simulated training system for observer training, in which fixed and rotary wing aircrafts, ground vehicles, and weapon effects are combined with real world scenes.

Index Terms—IMU, monocular wide and narrow field of view camera, GPS, inertial navigation, sensor fusion, EKF

1 Introduction

1.1 Motivation

Augmented reality (AR) slowly but steadily comes into our everyday life allowing more applications for professionals and ordinary consumers. Rapid increases in computer processing power and in the quality of cameras, combined with decrease in price and size of other sensors like inertial measurement unit (IMU) and global positioning system (GPS) allow quite compelling systems to be built. In this work we want to extend the augmented reality applications to telescopic systems, like binoculars. Availability of AR binoculars is very handy to improve the experience of the large-scale outdoor augmented reality scenarios, as distant augmentations that appear too small in the "unaided eye" can be interacted only via zooming optics. For example, even large entities like cars occupy only few pixels in the augmented image when inserted about a kilometer away.

Furthermore, in contrast to traditional augmented reality systems, AR binoculars are very natural to interact with. The device itself is contained in the binoculars shell of the standard size and does not require additional bulky inconvenient hardware in terms of helmet or special glasses, which makes it very attractive to training applications that have to use binoculars or other telescopic devices.

A good example of AR binoculars use would be military training of forward observers for the artillery and airstrike support missions, as depicted in Fig. 1. More broadly, AR binoculars can be used for training of military, security, disaster relief, fire fighters, and other surveillance applications. Finally, AR binoculars are suitable in consumer applications that involve bird watching, sport spectator-ship and hunting. In training applications, synthetic elements (vehicles, people, effects) are inserted into the binoculars' view to appear as if they are part of the live real world scene. Inserting virtual objects into the live scene saves the cost of using real vehicles, airplanes, actors, munitions etc. required for training. The inserted objects and effects must be realistic and must not jitter or drift when the user moves and pans around to visualize the scene from different locations and orientations.

Another application of AR binoculars is information sharing, where objects of interests are tagged and labeled in the augmented view. In this case, objects denoted by their 3D geo-spatial location also have information tags, labels and links associated with them. When a user looks in the direction of a visible object of interest, it should appear labeled in the scene. In this case too, the labels and tags must appear accurately in the augmented view. The labels should not jump from one object to another and should not jitter or drift as the user pans around.

Augmentation of highly zoomed images, as in binoculars, imposes strict constraints on the precision of 6 degree of freedom (DoF) tracking. In our case, the augmentation of 6.3 degree narrow field of view (FOV) color image of $1{,}280 \times 960$ resolution requires orientation estimation accuracy of around 0.005 degree in order to guarantee less than 1 pixel displacement error. That requires the development of robust and extremely accurate pose tracking system.

1.2 Previous Work

Augmented reality imposes a variety of stringent constraint for systems to be compelling and interactive. They must be real-time, have low latency and good precision. Consequently, researchers try to use virtually all possible sensors indicative of position and orientation in order to provide rapid and accurate pose estimation. An early work related to ours is that of [2], which used a differential GPS receiver, compass, gyros and tilt sensor on a static (non-portable)

• *The authors are with the Center for Vision Technologies, SRI International, Princeton, NJ 08540.*
E-mail: {taragay.oskiper, mikhail.sizintsev}@sri.com.

Manuscript received 1 Feb. 2014; revised 21 Feb. 2015; accepted 23 Feb. 2015. Date of publication 3 Mar. 2015; date of current version 1 Apr. 2015. Recommended for acceptance by M. Gandy, S. Julier, and K. Kiyokawa. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TVCG.2015.2408612

Fig. 1. Top row: Concept of Augmented Reality binoculars. Real scene is augmented with rendered objects (e.g., truck) and effects (e.g., explosion). Bottom row: Example augmented view of the normal "unaided eye" observer (left) and augmented view through the binoculars (right). Zoomed-in area is marked with yellow ellipse.

platform. The real-time system did not have a vision sensor, instead it combined rate gyros with a compass and tilt orientation sensor in a hybrid tracker. Large global drift due to compass errors required sensor recalibration, and local registration errors of 2 degrees were achieved. Subsequently, [11] described a 3DoF camera tracking system that employs a high precision gyroscope and a vision-based drift compensation algorithm by tracking natural features in the outdoor environment using a template matching technique for landmark detection. More recent work (e.g., [10], [21], [23]) concentrated on wide area augmented reality systems that rely on known models of the environment and require offline preparation stage. Yet another type of solution performs the tracking of detachable camera with respect to mobile platform that has odometry measurements of its own (e.g., [24]).

Importantly, there have been a number of purely vision based AR systems. While early and some recent systems concentrated on marker-based augmented reality for speed, simplicity and low hardware requirements (e.g., [6], [19], [20]), a number of good marker-less vision-only pose tracking systems that use natural image features/lines/edges appeared over the past decade (e.g., [4], [7], [12], [17]). Nowadays, vision-only systems come handy in cell phone applications (e.g., [1]), because even though mobile devices have gyroscopes and accelerometers, the quality of the inertial sensors and synchronization with camera is relatively poor. Consequently, their employment in AR application is an active ongoing research work and interesting solutions have been proposed. For example, authors of [14] use gravity direction rather than complete IMU readings to significantly aid vision-only tracking running on a cell phone.

Our approach is based on visual-inertial navigation as we combine IMU-sensed ego-motion corrected by visual information in terms of feature tracking as well as landmark and panorama matching to bound the drift of otherwise dead-reckoning system. For visual-inertial 6DoF pose estimation, we rely on our previous work [18] (which non-trivially extends the approach of [16] to work with IMU of inferior quality) and use two cameras with significantly different fields of view to operate in parallel.

The term "augmented reality binoculars" and the general idea of augmenting the zoomed views with extra information has been realized before. In one of the earlier examples, the developer of concept prototype[1] uses virtual binoculars, which is really a head-mounted display in a form factor of binoculars, and augments the image of wide FoV external camera using GPS and gyroscopes. In this prototype, only tags and text are overlayed, device itself must be stationary on a tripod and zoomed-in view is never directly augmented. Another example of AR binoculars is an installation in Zurich airport set up by Artcom[2]. It is a stationary calibrated binocular system which displays information about the planes in the view. Again, no attempt of precise registration of synthetic entities with real world is demonstrated and overall system is not mobile. A very similar product called Augmented Reality Binoculars is offered by Trillian[3]. It is meant to be installed for panorama observation, adding names and product information while scene is panning and mostly used as a playback station for visual content such as photos and films. In conclusion, we are not aware of any specific work which properly addresses the issues of telescopic augmented reality in terms of precise jitter-free insertion and in a form factor of a hand-held binoculars.

1.3 Contribution

In light of previous research and current objectives, this paper presents the following contributions. We present a novel design of Augmented Reality Binoculars. We develop a robust and accurate algorithm to track the 6-DoF pose of the AR binoculars using IMU, GPS, wide FoV and narrow FoV cameras fused in an error-state extended Kalman filter (EKF). In this system, wide FoV camera is used for robust general 6-DoF tracking, while narrow FoV camera is used for augmentation and 3-DoF pose tracking refinement. Furthermore, visual landmark matching and panorama mechanism allow for rapid global correction of orientation and minimize the drift in orientation estimation which is inevitable in any dead-reckoning system that uses IMU and cameras only. The method is currently implemented and runs on a laptop platform with very low latency while quantitative and qualitative experimental evaluation shows the applicability and versatility of the proposed AR binoculars.

2 Technical Approach

2.1 AR Binocular System and Hardware

We design our augmented reality binoculars taking the Vector-21 device by Vectronix[4] as a prototype, currently tethered to a laptop. Its appearance and internal components

1. http://www.jarrellpair.com/augmented-reality-binoculars
2. http://www.artcom.de/en/projects/project/detail/visitor-deck-dock-b/
3. http://www.trillian.de/en/fernglaeserI.htm
4. http://www.vectronix.com

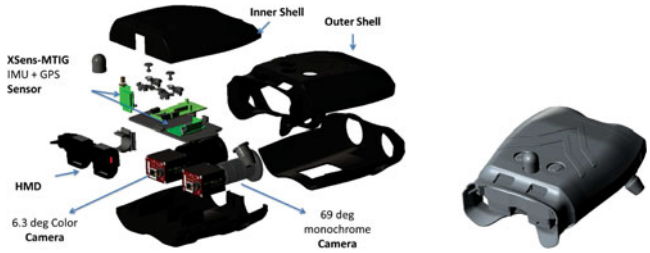


Fig. 2. The design and appearance of our AR binoculars. Our system consists of XSens IMU and GPS package, wide FoV monochrome camera for robust 6-DoF pose tracking, narrow FoV color camera for augmentation and increased precision in tracking, HMD for display.

are shown in Fig. 2. The sensor package consists of two cameras (narrow and wide FoV) and an XSens-MTi-G[5] which includes MEMs type IMU, GPS, magnetometer and barometric pressure sensor. The IMU and cameras are synchronized using the built-in XSens trigger (Xsens sync-out is used to trigger the cameras). We have developed two types of the binocular hardware. In the first version, the zoomed-in view is captured by a color $1{,}280 \times 960$ Prosilica camera[6] with 6.3 degree (7x) FoV lens and in the the second setup, same type of camera with 4.725 degree (10x) FoV lens is used. The images from the narrow FoV camera are used for augmentation and to add precision to the tracking. Due to its very narrow FoV, it is not possible to perform reliable visual tracking with it alone; thus a second monochrome $1{,}280 \times 960$ Prosilica camera with a much larger field of view is introduced to aid in visual-inertial tracking. Section 2.4 describes the actual cooperation of two cameras in tracking. Conveniently, both cameras reside in the lens bay of the binoculars, while optional antenna can be externally attached for better GPS reception. Finally, the visual output is supplied by a removable head-mounted display (HMD) (in particular, we experimented with Vuzix Wrap 920 HMD[7] and Intevac I-Port 75[8] eyewear).

2.2 Coordinate Systems Description and Sensor Calibration

In Fig. 3, the relevant coordinate systems used in our extended Kalman filter solution are illustrated. The EKF provides estimates of the ground (global coordinate frame) to IMU pose, denoted as $\mathbf{P}_{GI} = [\mathbf{R}_{GI} \quad \mathbf{T}_{GI}]$. In this representation, a point $\mathbf{X}_G$ expressed in the ground frame can be transferred to the IMU coordinates by $\mathbf{X}_I = \mathbf{R}_{GI}\mathbf{X}_G + \mathbf{T}_{GI}$. Accordingly, $\mathbf{T}_{GI}$ represents the ground origin expressed in the IMU coordinate frame, whereas $\mathbf{T}_{IG} = -\mathbf{R}_{GI}^T\mathbf{T}_{GI}$ is the location of the IMU in the ground coordinate frame. The final output of the system is the ground to narrow FoV camera pose which is determined by the relation $\mathbf{P}_{GC^n} = [\mathbf{R}_{IC^n}\mathbf{R}_{GI} \; \mathbf{R}_{IC^n}\mathbf{T}_{GI} + \mathbf{T}_{IC^n}]$. For our method, we assume both the wide and narrow FoV camera intrinsics, and all the extrinsics between the IMU and cameras are known.

In order to determine the fixed relation between the IMU and the wide FoV camera coordinate systems, which we refer as the IMU to camera pose, $\mathbf{P}_{IC^w} = [\mathbf{R}_{IC^w} \; \mathbf{T}_{IC^w}]$, we use an extrinsic calibration procedure, as developed in [15]. The intrinsics of the wide FoV camera are obtained using a checkerboard pattern and MATLAB calibration toolbox[9]. However, this method is very challenging for the narrow FoV camera intrinsic calibration since it would require either a very large checkerboard to be placed at a far distance or very small pattern placed close to the camera which would be problematic to focus properly. Furthermore, performing the narrow-to-wide extrinsic calibration using standard two-camera stereo-like calibration techniques is hardly applicable, since designing and simultaneously observing a calibration pattern with cameras having 10 to 16-fold FoV difference would also be rather impractical. Also, the IMU to camera extrinsic calibration method [15] that we used for the wide FoV camera is not suitable for the narrow FoV camera since this particular method not only requires the camera intrinsics to be known, but it also depends on the user moving the sensor rig in front of a checkerboard in order to generate motion estimates from both IMU and the camera. Instead we developed an effective method to obtain the narrow FoV camera's intrinsic and extrinsic calibration parameters, which is based on the idea of calibrating narrow FoV camera relative to the wide FoV camera using natural scene.

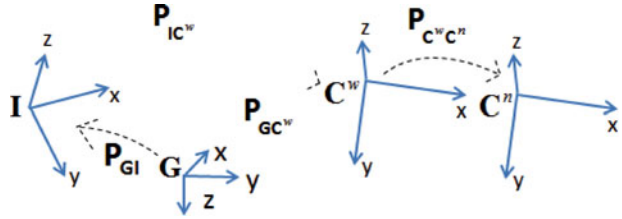


Fig. 3. Relation between the global (ground), IMU, wide FoV and narrow FoV camera coordinate frames (denoted by the letters $\mathbf{G}$, $\mathbf{I}$, $\mathbf{C}^w$, and $\mathbf{C}^n$ respectively) are shown. Vertical direction, depicted by the z-axis of the ground frame (in the north-east-down, NED, convention), points in the direction of gravity. IMU and cameras are rigidly connected, whose extrinsics are stored in the IMU to camera pose matrices $\mathbf{P}_{IC^w}$ and $\mathbf{P}_{IC^n} = \mathbf{P}_{C^wC^n}\mathbf{P}_{IC^w}$. The system tracks the ground to IMU pose, $\mathbf{P}_{GI}$, which together with $\mathbf{P}_{IC^n}$, is used in determining the ground to narrow FoV camera pose, $\mathbf{P}_{GC^n}$, as the final system output.

We start by making the following three assumptions. First, the radial distortion of the narrow FoV camera is negligible and can be set to zero. Second, the principal point coincides with the image center and the horizontal and vertical focal lengths are equal, i.e. intrinsic calibration matrix $\mathbf{K}_{C^n}$ is defined just by the focal length f_{C^n}:

$$\mathbf{K}_{C^n} = \begin{bmatrix} f_{C^n} & 0 & \text{width}/2 \\ 0 & f_{C^n} & \text{height}/2 \\ 0 & 0 & 1 \end{bmatrix}. \tag{1}$$

Another reason to rely on this assumption is narrow FoV principal point coordinates are interdependent with rotation matrix x- and y-axes components between narrow and wide FoV cameras. And third, we assume the translation between the wide and narrow cameras can be ignored such that $\mathbf{P}_{C^wC^n} = \mathbf{R}_{C^wC^n}$, due to the difference in resolution and because we normally look at distant objects.

5. http://www.xsens.com/en/general/mti-g
6. http://www.alliedvisiontec.com
7. http://www.vuzix.com
8. http://www.intevac.com/intevacphotonics/vision-systems/vision-systems-products/i-port/i-port-systems/
9. http://www.vision.caltech.edu/bouguetj/calib_doc

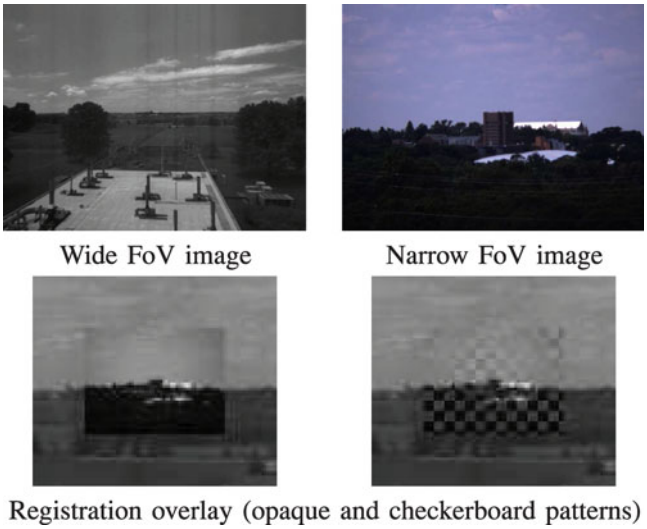


Fig. 4. The narrow-to-wide camera calibration is performed by registering narrow FoV image to the corresponding central portion of the wide FoV image. The warp map is used to optimize for relative orientation of the narrow FoV camera to the wide FoV camera as well as its focal length. The combined image shows central cutout of the wide FoV image with registered narrow FoV image to demonstrate the alignment.

Thus, our goal is to find the rotation matrix $\mathbf{R}_{C^wC^n}$, which can then be used to obtain $\mathbf{P}_{IC^n} = [\mathbf{R}_{C^wC^n}\mathbf{R}_{IC^w} \quad \mathbf{0}]$, and the unknown focal length of the narrow FoV camera f_{C^n}. Considering the set of corresponding points between wide and narrow camera, $\mathbf{p}_j^w$ and $\mathbf{p}_j^n$, resp., the objective is to minimize the Euclidean image distance in pixels

$$E = \sum_j \left\| \mathbf{p}_j^w - h\left(\mathbf{K}_{C^w}\mathbf{R}_{C^nC^w}\mathbf{K}_{C^n}^{-1} \begin{bmatrix} \mathbf{p}_j^n \\ 1 \end{bmatrix} \right) \right\|^2, \qquad (2)$$

where $h(x) = [\, x_1/x_3 \quad x_2/x_3 \,]^\top$ stands for the pin-hole perspective projection. The above problem (2) is solved via standard Levenberg-Marquardt procedure [9]. However, the important requirement is to determine the set of correspondences that comprise its inputs. In our case, it is particularly difficult to find features and determine correspondences due to extremely large resolution difference in two images. Instead, we determine dense correspondences; specifically, we register the narrow and wide FoV images using an affine model and Lucas-Kanade method performed over Laplacian pyramid to compensate for radiometric differences [3]. Initialization is done by reducing the narrow FoV image by a factor so that common scene features appear roughly similar size in pixels for both images and placing it in the center of the wide FoV image.

Fig. 4 visualizes the result of calibration between wide and narrow FoV images. Original images and the narrow FoV overlayed on wide FoV according to the homography, $\mathbf{H}_{C^wC^n} = \mathbf{K}_{C^w}\mathbf{R}_{C^wC^n}\mathbf{K}_{C^n}^{-1}$, determined after solving (2) is shown using a checkerboard pattern to demonstrate the quality of alignment.

2.3 Extended Kalman Filter Process Model

We build on our previous work [18] in which we introduced an error-state EKF that performs sensor fusion between a monocular camera and an IMU for visual-inertial odometry.

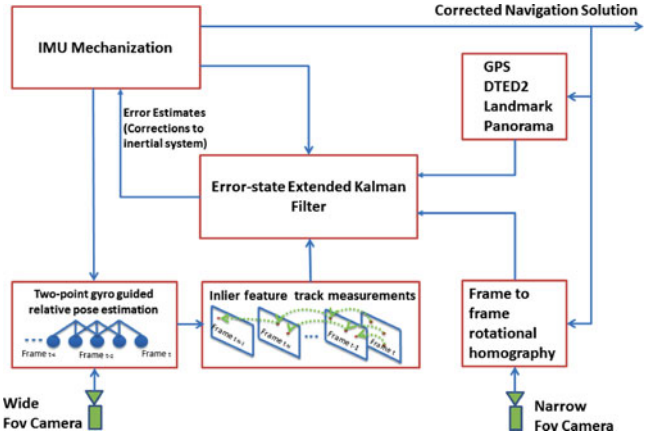


Fig. 5. System block diagram for pose tracking.

The filter provides 6-DoF pose estimates for navigation by generating relative visual measurements at the feature track level and marginalizing out the 3D feature points, obtained via multi-view triangulation, from the measurement model. This reduces the state vector size and makes real time implementation possible by keeping computational complexity linear in the number of features. In this work, we expand the algorithms and functionality in order to incorporate two cameras (both employed in monocular fashion) and additional global measurements in the form of matches to a panorama database which is created online. The overall system block diagram for pose tracking is depicted in Fig. 5.

The total (full) states of the EKF consist of the IMU location $\mathbf{T}_{IG}$, the gyroscope bias vector $\mathbf{b}_g$, velocity vector $\mathbf{v}_{IG}$ in global coordinate frame, accelerometer bias vector $\mathbf{b}_a$ and ground to IMU orientation $\mathbf{q}_{GI}$, expressed in terms of the quaternion representation for rotation, such that $\mathbf{R}_{GI}(\mathbf{q}_{GI}) = (|q_0|^2 - \|\vec{\mathbf{q}}\|^2)\mathbf{I}_{3\times3} + 2\vec{\mathbf{q}}\,\vec{\mathbf{q}}^T - 2q_0[\vec{\mathbf{q}}]_\times$, with $\mathbf{q}_{GI} = [q_0 \quad \vec{\mathbf{q}}^T]^T$ and $[\vec{\mathbf{q}}]_\times$ denoting the skew-symmetric matrix formed by $\vec{\mathbf{q}}$. 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

$$\mathbf{s} = \begin{bmatrix} \mathbf{q}_{GI}^T & \mathbf{b}_g^T & \mathbf{v}_{IG}^T & \mathbf{b}_a^T & \mathbf{T}_{IG}^T \end{bmatrix}^T. \qquad (3)$$

Accordingly, we use the same state time evolution and IMU mechanization model as in [18]. The 15 dimensional Kalman filter error state consists of

$$\delta\mathbf{s} = \begin{bmatrix} \delta\Theta_G^T & \delta\mathbf{b}_g^T & \delta\mathbf{v}_G^T & \delta\mathbf{b}_a^T & \delta\mathbf{T}_G^T \end{bmatrix}^T, \qquad (4)$$

which is based on the following relation between the total state and its inertial estimate:

$$\mathbf{q}_{GI} = \delta\mathbf{q}_G \otimes \hat{\mathbf{q}}_{GI}, \quad \text{with} \quad \delta\mathbf{q}_G \simeq \begin{bmatrix} 1 & \delta\Theta_G^T/2 \end{bmatrix}^T, \qquad (5)$$

$$\mathbf{b}_g = \hat{\mathbf{b}}_g + \delta\mathbf{b}_g, \ \mathbf{b}_a = \hat{\mathbf{b}}_a + \delta\mathbf{b}_a, \qquad (6)$$

$$\mathbf{v}_{IG} = \hat{\mathbf{v}}_{IG} + \delta\mathbf{v}_G, \ \mathbf{T}_{IG} = \hat{\mathbf{T}}_{IG} + \delta\mathbf{T}_G\,. \qquad (7)$$

Finally, the dynamic process model for the error state is given by

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

where

$$\mathbf{F} = \begin{bmatrix} \mathbf{0}_{3\times3} & -\hat{\mathbf{R}}_{GI}^{T} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ -\left[\hat{\mathbf{R}}_{GI}^{T}\hat{\boldsymbol{\alpha}}\right]_{\times} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\hat{\mathbf{R}}_{GI}^{T} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix},$$

$$\mathbf{n} = \begin{bmatrix} \mathbf{n}_g \\ \mathbf{n}_{wg} \\ \mathbf{n}_a \\ \mathbf{n}_{wa} \end{bmatrix}, \text{ and } \mathbf{G} = \begin{bmatrix} -\hat{\mathbf{R}}_{\mathrm{GI}}^{\mathrm{T}} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\hat{\mathbf{R}}_{\mathrm{GI}}^{\mathrm{T}} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}.$$

The noise term $\mathbf{n}$ and acceleration term $\hat{\boldsymbol{\alpha}}$ above are the same as in [18]. However, note that the parametrization of the orientation error in (5) is different than the one used in [18]—left-sided multiplicative error term versus right-sided error term in our quaternion notation. As such, the state transition matrix $\mathbf{F}$ and noise matrix $\mathbf{G}$ are different than those of [18]. According to this new representation, the relation between the true and the estimated orientation is given by

$$\mathbf{R}_{GI} = \hat{\mathbf{R}}_{GI}\ e^{-[\delta\Theta_G]_{\times}} \simeq \hat{\mathbf{R}}_{GI}\left(\mathbf{I}_{3\times3} - [\delta\Theta_G]_{\times}\right) \ . \tag{9}$$

(Note that the multiplicative error term is on the right hand side in the rotation matrix representation above. This is due to fact that, given a sequence of rotations represented by the quaternions $\mathbf{q}_1$ and $\mathbf{q}_2$, we have the relation $\mathbf{q}_1 \otimes \mathbf{q}_2 = \mathbf{R}(\mathbf{q}_2)\mathbf{R}(\mathbf{q}_1)$ based on our quaternion representation.) Hence, orientation error is the error of the world coordinate frame orientation as opposed to the IMU coordinate frame orientation and therefore expressed in the global world coordinate frame as opposed to the local IMU coordinate frame. Accordingly, orientation uncertainty is expressed in the global coordinate frame regardless of the current IMU orientation and this representation allows for easier interpretation of the orientation error covariances as we refer in the next section.

2.4 Multi-camera Feature Tracking and Odometry

As the video frames from both narrow and wide FoV cameras are received, feature extraction and frame to frame feature matching is performed simultaneously in both cameras as depicted in Fig. 6. In order to save computational resources however, the narrow FoV feature tracking is activated only during small camera motion (when rotational velocity is below a certain threshold, set at 30 degrees/sec in our case, for which there is sufficient overlap between frames). The wide FoV camera measurements are incorporated into the EKF by using the same feature track based relative measurement model as described in [18], after the Jacobians are modified to reflect the new parametrization of the orientation error state as expressed in the global frame instead of the local IMU coordinate frame. As for the narrow FoV camera, a measurement model based only on frame-to-frame relative rotation is established as we will show in this section.

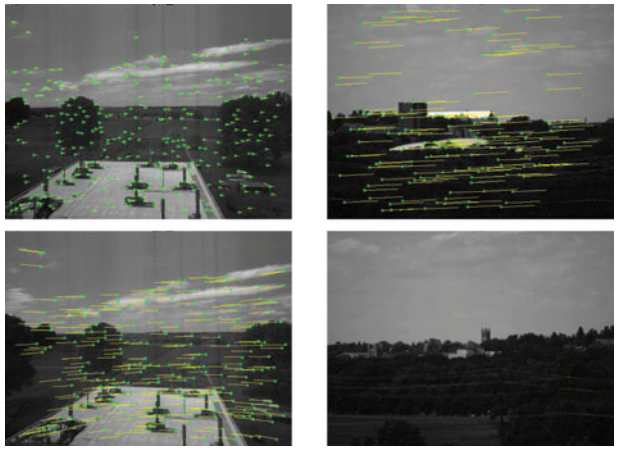


Fig. 6. Inlier feature tracks obtained from wide and narrow FoV cameras. At the top row, two frames extracted at the same time from wide and narrow FoV cameras are shown overlaid with the feature track information, with tracks extended one frame into the past as shown in yellow lines. Small camera motion allows successful feature tracking on both cameras, whereas the bottom row demonstrates the feature tracking failure in the narrow FoV camera with moderate camera motion (bottom right). In this situation, the system automatically relies on the wide FoV camera for visual odometry measurements (bottom left). See supplementary video, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2015.2408612.

Measurements from the narrow FoV camera become available whenever we detect N inlier feature correspondences between two narrow FoV camera views, which are determined by thresholding around the EKF predicted feature locations and correlation matching amongst the candidates, (in our case N is set to 20, so if there are less than 20 inliers we do not proceed with the following). Letting $\mathbf{x}_{C_1^n}^k$ and $\mathbf{x}_{C_2^n}^k$ be the normalized image coordinates for the k'th such correspondence (having removed the camera intrinsics), we solve for the relative rotation $\mathbf{R}_{C_1^n C_2^n}$ that minimizes the sum of norm-squared errors:

$$E = \Sigma_{k=1}^{N} \left\| \boldsymbol{\epsilon}^k \right\|^2, \quad \text{where} \quad \boldsymbol{\epsilon}^k = \mathbf{x}_{C_2^n}^k - h(\mathbf{y}^k) \tag{10}$$

and

$$\mathbf{y}^k = \mathbf{R}_{C_1^n C_2^n} \begin{bmatrix} \mathbf{x}_{C_1^n}^k \\ 1 \end{bmatrix} \quad \text{with} \quad h(\mathbf{y}^k) = [y_1^k/y_3^k \ \ y_2^k/y_3^k]^T \ . \tag{11}$$

We solve the above least squares problem via iterative nonlinear minimization starting with the EKF predicted relative orientation and ending after two iterations. If we let $\widetilde{\mathbf{R}}_{C_1^n C_2^n}$ denote the attained solution, and $\widetilde{\mathbf{y}}^k = \widetilde{\mathbf{R}}_{C_1^n C_2^n}\, \mathbf{x}_{C_1^n}^k$, then the corresponding Jacobians at the solution point are given by

$$\mathbf{J}_{\boldsymbol{\epsilon}}^k = \begin{bmatrix} 1/\widetilde{y}_3^k & 0 & -\widetilde{y}_1^k/(\widetilde{y}_3^k)^2 \\ 0 & 1/\widetilde{y}_3^k & -\widetilde{y}_2^k/(\widetilde{y}_3^k)^2 \end{bmatrix} [\widetilde{\mathbf{y}}^k]_{\times} \ . \tag{12}$$

Using back propagation of covariance [9], the uncertainty corresponding to this relative orientation estimate is given by

$$\Sigma_{\eta_C} = \left(\Sigma_{k=1}^{N} \mathbf{J}_{\boldsymbol{\epsilon}}^{k\,T} \mathbf{J}_{\boldsymbol{\epsilon}}^k\right)^{-1} \sigma_p^2 \ , \tag{13}$$

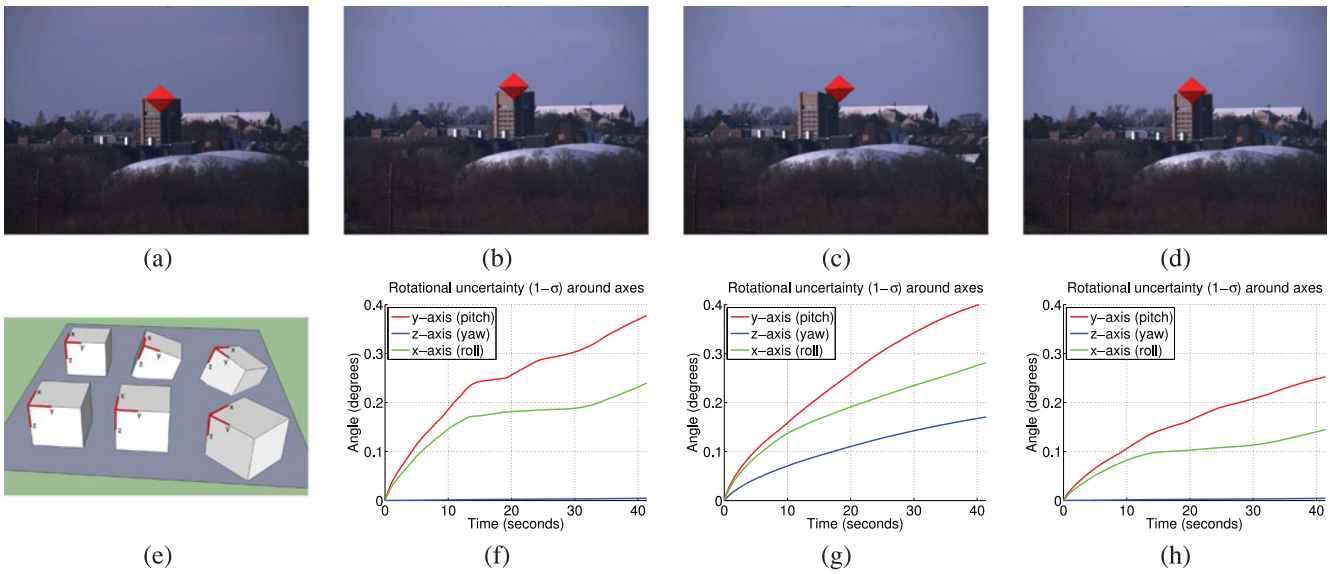


Fig. 7. Demonstration of dead-reckoning performance of sensor fusion using odometry measurements from wide and narrow FoV cameras on a short sequence where the binoculars are panned left and right on a tripod. Inset (a) shows the marker position at the start of the sequence, and insets (b), (c), and (d) show the marker positions at the end of the sequence when either only the narrow FoV camera is used, or only the wide FoV camera is used, or both narrow and wide FoV cameras are used, respectively. Insets (f), (g), and (h), show the corresponding uncertainty growth in orientation as a function of time for the cases of narrow FoV only, wide FoV only, and narrow + wide FoV , respectively. Inset (e) illustrates how drift in relative roll estimates cause global drift in both roll and pitch.

where σ_p is the standard deviation for the feature localization uncertainty in normalized units, which we set at $1/f_{C^n}$ (c.f. (1)), corresponding to 1 pixel std in our case. Based on this and the IMU to narrow FoV camera extrinsics, the relative orientation measurement for the IMU between two time instances is given by

$$\widetilde{\mathbf{R}}_{I_1I_2} = \mathbf{R}_{IC^n}^T \widetilde{\mathbf{R}}_{C_1^nC_2^n} \mathbf{R}_{IC^n} \quad \text{and} \quad \Sigma_{\eta_I} = \mathbf{R}_{IC^n}^T \Sigma_{\eta_C} \mathbf{R}_{IC^n}, \tag{14}$$

where $\Sigma_{\boldsymbol{\eta}_I}$ is the measurement noise covariance transformed from camera to IMU coordinate frame.

Next, if we write the true relative rotation for the IMU in terms of the IMU orientation at the previous and current time instances,

$$\mathbf{R}_{I_1I_2} = \mathbf{R}_{GI_2} \mathbf{R}_{GI_1}^T , \tag{15}$$

we obtain under small error assumption

$$\begin{aligned} \hat{\mathbf{R}}_{I_1I_2} \left(\mathbf{I}_{3\times3} - \left[\delta\Theta_{I_{1,2}}\right]_\times\right) \simeq \hat{\mathbf{R}}_{GI_2} \left(\mathbf{I}_{3\times3} - \left[\delta\Theta_{G_2}\right]_\times\right) \\ \left(\mathbf{I}_{3\times3} + \left[\delta\Theta_{G_1}\right]_\times\right) \hat{\mathbf{R}}_{GI_1}^T , \end{aligned} \tag{16}$$

$$\hat{\mathbf{R}}_{I_1I_2} \left[\delta\Theta_{I_{1,2}}\right]_\times \simeq \hat{\mathbf{R}}_{GI_2} \left(\left[\delta\Theta_{G_2}\right]_\times - \left[\delta\Theta_{G_1}\right]_\times \right) \hat{\mathbf{R}}_{GI_1}^T . \tag{17}$$

From this we get,

$$\delta\Theta_{I_{1,2}} \simeq \hat{\mathbf{R}}_{GI_1} \delta\Theta_{G_2} - \hat{\mathbf{R}}_{GI_1} \delta\Theta_{G_1}, \tag{18}$$

where we used the fact that for a given 3×3 orthonormal matrix $\mathbf{A}$ and a 3×1 vector $\mathbf{b}$, $\mathbf{A}[\mathbf{b}]_\times = [\mathbf{A}\mathbf{b}]_\times \mathbf{A}$. The above relation describes the relative orientation error in terms of the orientation error state corresponding to the previous and current time instances. The relative measurement residual, $\delta\mathbf{z}$, between the two time instances is obtained in terms of the rotational difference between the EKF estimate and the measurement, $\widetilde{\mathbf{R}}_{I_1I_2}$, as found in (14). Accordingly,

$$\widetilde{\mathbf{R}}_{I_{1,2}} = \hat{\mathbf{R}}_{I_1I_2} e^{-[\delta\mathbf{z}]_\times} \quad \text{and} \quad \delta\mathbf{z} \simeq \delta\Theta_{\mathrm{I}_{1,2}} + \boldsymbol{\eta}_{\mathrm{I}} \tag{19}$$

with Σ_{η_I} given in (14). Finally, the measurement residual can be expressed in terms of the previous and current filter states in the following fashion:

$$\delta\mathbf{z} \simeq \mathbf{H}_1 \delta\mathbf{s}_1 + \mathbf{H}_2 \delta\mathbf{s}_2 + \eta_I \tag{20}$$

with the measurement Jacobians

$$\mathbf{H}_1 = [-\hat{\mathbf{R}}_{GI_1} \quad \mathbf{0}_{3\times12}], \quad \mathbf{H}_2 = [\hat{\mathbf{R}}_{GI_1} \quad \mathbf{0}_{3\times12}], \tag{21}$$

where we used (18), and the fact that error-state vectors $\delta\mathbf{s}_1$ and $\delta\mathbf{s}_2$ are 15 dimensional (c.f. (4)). Hence, the filter updates corresponding to this measurement model can be performed via stochastic cloning [22] by augmenting the current state with a copy of the previous state.

In Fig. 7, we demonstrate on a short sequence how the filter fuses visual odometry measurements from the narrow and wide FoV cameras. For this purpose, the binoculars are placed on a tripod and panned slowly leftward for about 20 degrees and then panned back to the beginning position. Successful frame-to-frame feature tracking is maintained for both the wide and narrow FoV cameras during the whole time so that the filter receives relative motion measurements from both. Fig. 7a shows the marker position at the beginning of the sequence (right after the initialization which is described in Section 2.5). Figs. 7b, 7c, and 7d show the marker positions at the end of the sequence when either only the narrow FoV camera is used, or only the wide FoV camera is used, or both narrow and wide FoV cameras are used, respectively.

Focusing on the narrow FoV case for the time being, from Fig. 7b one can see that, at the end of the sequence, global yaw is very accurate as the marker appears in the correct spot horizontally, however drift in global pitch is noticeable. The reason for this is the following. Narrow FoV camera

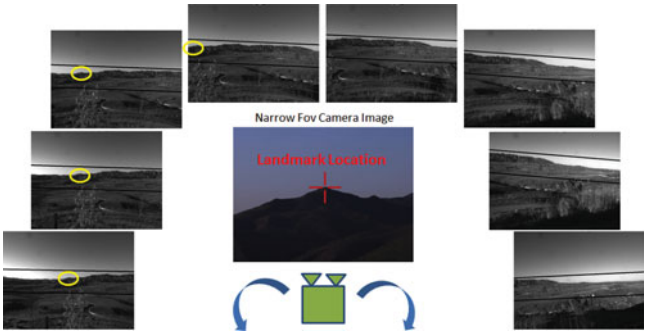


Fig. 8. System initialization procedure followed by online panorama generation. After system start-up, the user roughly aligns the cross-hair in the video frame with the known 3D landmark in the environment, at which point the system automatically matches the current narrow FoV video frame to the landmark database image to register the 2D pixel location of this landmark in the query image. Later on, as the user pans the camera left and right during the exercise, the system automatically extracts overlapping frames and creates a panoramic database of the area of interest covered up to that moment. Yellow oval shows the location of initial landmark and black lines show the horizon band features which are used in panorama database construction and query matching.

allows very high precision in terms of feature localization. On the other hand, due to its nature, relative rotation measurements obtained from the narrow FoV camera exhibit high uncertainty in roll. Intuitively, as the FoV of a camera gets smaller, the rotations around its principal axis (roll-axis) get harder to infer, since camera motion around this axis creates far less image motion than similar amount of camera motion around either the pitch or yaw axes. For instance, in this sequence, average values for the square root of the diagonal terms of the noise covariance matrix obtained according to (13) are [0.0006 0.0006 0.02] degrees. As expressed in the camera coordinate frame, these values suggest much higher uncertainty for rotations around the z-axis of the camera. (Note that whereas the z-axis of the camera frame stands for its roll axis, in the case of the global coordinate frame, roll is indicated by rotations around the x-axis, c.f., Fig. 3). As a consequence, error in relative roll measurements cause drift in both global pitch and global roll orientation which is illustrated in Fig. 7e. The bottom row in this figure (from left to right) shows the true orientation of a cube at three different time instances t_1, t_2 and t_3. Between t_1 and t_2 the cube undergoes no rotation, and between t_2 and t_3 a 45 degree rotation around the vertical axis (yaw-axis) is applied. The top row shows the estimated orientation of the cube, provided by a system that introduces a 20 degree roll error (which is quite large for illustration purposes) in its estimate of the relative rotation between t_1 and t_2 and otherwise perfectly estimates the relative yaw and pitch between t_1 and t_2 and between t_2 and t_3. At the end of t_3, the large drift in pitch is easily visible.

In Figs. 7f, 7g, and 7h we plot the square root of the first three diagonal terms of the EKF error-state covariance matrix that correspond to the filter's estimate of the orientation error standard deviation as a function of time. As the error-state for orientation is expressed in the global coordinate frame, c.f. (9), the interpretation of rotational uncertainties around each world axis (in terms of roll, pitch and yaw) is readily available regardless of the IMU orientation. It can bee seen from Fig. 7f that, in the case of the narrow FoV camera, whereas the global yaw uncertainty increases at a very low rate, the global pitch and roll uncertainties increase at a much faster rate. In the case of the wide FoV camera only, uncertainties in orientation around all axes increase quite rapidly (Fig. 7g), and the drift in yaw as well as pitch, is easily visible (Fig. 7c). Finally, according to Fig. 7h), the combined system exhibits uncertainty growth around the yaw-axis equivalent to the narrow FoV only case, and a noticeably lower rate of increase in uncertainty around the pitch and roll axes. The marker location depicted in Fig. 7d is in agreement with this observation and indicates very low drift in yaw and around half the drift in pitch as compared to the narrow FoV alone.

2.5 Initialization and Global Measurements

In order to determine the global orientation of the device after start-up, we use a landmark based initialization procedure similar to [18], where a 3D to 2D tiepoint is established between an image frame and a world location. We assume geodetic coordinates of the initial location and the coordinates for an easily discernible landmark in the scene are known. Also, similar to [18] a locally tangent plane coordinate frame is established at the initial location and used during the rest of the exercise. Geodetic coordinates of the landmark location are also transferred to this Euclidean coordinate frame.

After the landmark tiepoint click, the EKF is initialized and starts providing pose estimates. Whenever this single landmark tiepoint is in the field of view, the filter receives global orientation fixes and any drift in yaw is essentially reset. Importantly, the explicit click has to be done only once when the operator starts working in the new location. Once the image of the landmark has been captured, it can be reused in subsequent initializations for later use of the binocs and automatically matched to the incoming images for global yaw correction.

In order to increase the coverage of the system where global orientation measurements are available and avoid the need to designate multiple distinct landmarks with known 3D geodetic points in the environment, we have developed automatic online landmark expansion procedure. This improves the portability and set-up requirements of our system in addition to relieving the user from having to define multiple landmarks. After the initial 3D-2D landmark tiepoint matching is established, as the user pans around the binoculars, video frames from the wide FoV camera are collected to build an online panorama database (c.f. Fig. 8). A new image is added to the database if its overlap with current mosaic is more than a certain threshold. Matching of query to panorama database image is performed using Harris corner interest points [8] with Histogram of Gradients (HOG) descriptors [5]. Furthermore, for each received query image, the database is searched starting with the closest image in viewpoint to the query frame. The relative camera orientation between the query and database image is determined by solving the rotational homography in a robust RANSAC framework, as depicted in Fig. 9. After this step, the returned relative orientation is combined with the absolute orientation of the database frame, resulting in a global orientation measurement for the query frame. Importantly, we only consider interest points near the horizon band (defined by acceptable threshold on pitch estimates provided by the filter) as visualized by black lines in Figs. 8 and 9 because sky pattern above horizon line tends to

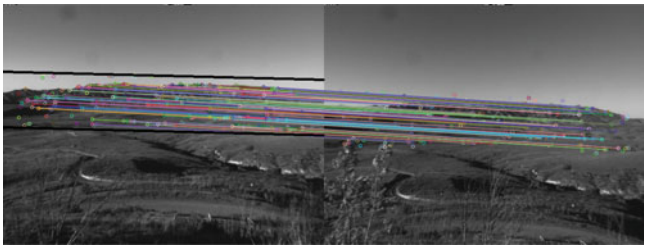


Fig. 9. Inlier matches between the panorama database image (right side) that is found as closest to the viewpoint of the query image (left side) are shown. Based on the current camera roll and pitch estimate fed by the EKF, region of interest around the horizon is determined for extracting the query features (shown by two black lines in the query image).

continuously change over time, while close objects that normally reside below horizon line can cause large visual parallax while moving the binoculars. Finally, during execution, all the camera orientations in the constantly growing panorama are refined using a bundle adjustment process [9] running on a separate thread.

Just as in the 3D-2D landmark tiepoint matching described in [18], panorama matching process typically takes few frames, hence these global orientation measurements arrive in a delayed manner to the EKF. In order to use these measurements correctly, the same procedure as the tiepoint landmark measurement is applied, so that they are propagated to the current frame time. This is done by combining the returned global orientation that belongs to a past query frame, with the relative orientation between the query frame and the current frame, for which the drift is assumed negligible.

In our filter framework, if we let $\widetilde{\mathbf{R}}_{GC^w}$ be the ground to wide FoV camera orientation measurement for the current time instant obtained in the above fashion, then using the camera to IMU extrinsics, we get $\widetilde{\mathbf{R}}_{GI} = \mathbf{R}_{C^wI}\widetilde{\mathbf{R}}_{GC^w}$ as the ground to IMU orientation measurement. Then if we let $\hat{\mathbf{R}}_{GI}$ be the predicted ground to IMU orientation for the current time instant, the measurement model in the error-states is expressed as

$$\delta\mathbf{z}_\theta = \delta\Theta_G + \gamma, \tag{22}$$

with the measurement residual obtained via $e^{-[\delta\mathbf{z}_\theta]_\times} = \hat{\mathbf{R}}_{GI}^T\widetilde{\mathbf{R}}_{GI}$ and γ is the noise with covariance Σ_γ.

2.6 Mobile AR Binoculars

Up to now the described system worked optimally around some Geo-Location where initialization procedure took place. In order to support a mobile platform and maintain long duration global heading accuracy, we need to expand our single panorama based framework which we use to match query images and receive orientation updates. In our current approach, we have implemented tools to easily build a set of panoramas (each indexed by its location at which it was built) and modified our landmark matching pipeline to efficiently query against this multiple panorama database to obtain global heading measurements into the EKF. During the database collection and generation process, we offer a Google Earth[10] like map interface where the user may select any one of the predetermined landmarks visible from his current location in the exercise area and record a panorama by panning around. These predetermined landmarks, stored with their associated geodetic coordinates, are chosen purely based on their saliency and the user's ability to distinctly identify them in the scenery. The top of Fig. 18 depicts prototype inferface and example of multiple panoramas. At the end of each such panorama the user is presented with an image of the first panorama frame where he can click on the corresponding landmark point to establish a tie between the geodetic landmark coordinate provided by the map interface and its corresponding image location. This information is subsequently used to compute and store the associated global heading for each panorama in the database. Afterward, this prebuilt database of panorama sets is loaded at the beginning of the actual exercise and when the user moves from one location to another, the corresponding nearest panorama is activated to be matched against the query frames. If the user moves to an observation location outside the coverage of this prebuilt set of panoramas, the system brings up the map interface where the user can select a landmark location and create a new panorama which is automatically added to the existing database.

So far in this framework, we have made the assumption that visual inertial odometry component of our pose estimation framework in the binocular platform is active at all times except for temporary outages due to partial or full occlusion by nearby objects or textureless regions such as sky, etc. However, in more realistic scenarios, the user is likely to carry the binoculars attached to a strap around the neck except for durations when they are raised to visualize the area of exercise. This is especially true while the user is on the move transitioning from one observation spot to another. During these periods, unaided-eye platform is expected to be utilized, while the binoculars are kept pointing down with visual tracking severely inhibited.

3 Experimental Evaluation

The system is implemented in C++ and executed on a Dell Precision M4600 with Intel Core i7-2820QM CPU and 4 GB of RAM. Average filter epoch (including feature tracking front end) computation takes about 30 milliseconds, which is more than acceptable for our 25 Hz video streaming even without the need for IMU-based pose predictions to eliminate any latency. Our experimental evaluation consists of two parts: quantitative and qualitative. First part will demonstrate the insertion of an octahedron marker at user-defined geo-locations and evaluation of insertion accuracy in terms of angular error in comparison to hand labeled ground truth. Second part will demonstrate the actual augmented reality military training application with stable insertions of vehicles (trucks, tanks and aircrafts) together with fire and explosion effects.

3.1 Information Sharing Application Example and System Performance

For the application of AR binoculars in information sharing, objects of interests are tagged and labeled in the augmented binocular view and tagged locations are shared between multiple AR devices at different places. Fig. 10 shows the marker labeling of three far distinct buildings seen from the

10. http://www.google.com/earth/index.html

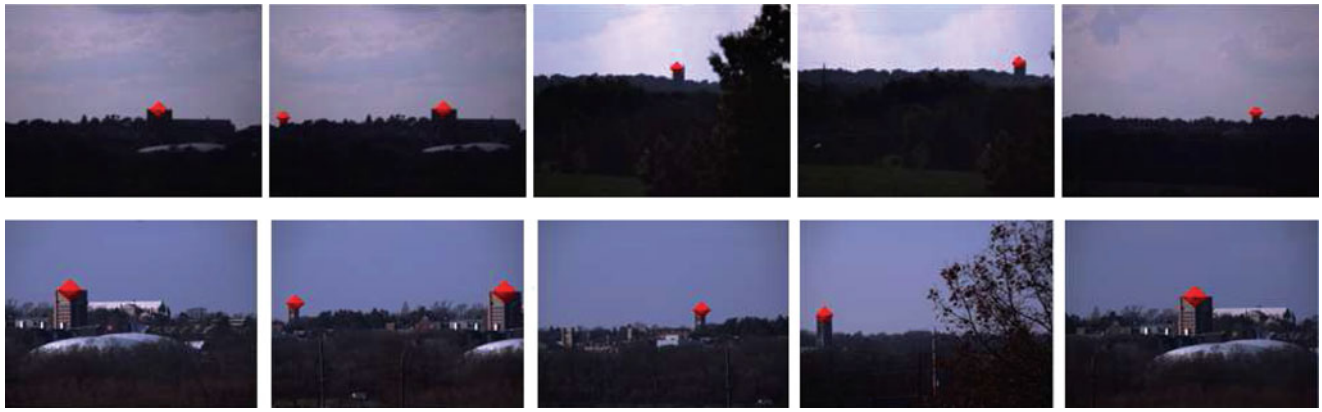


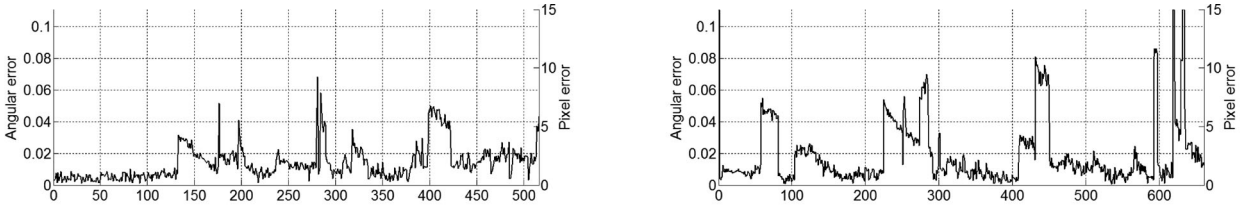
Fig. 10. Augmented reality example with geo-landmarks inserted at the location of three tallest university campus buildings observable via binoculars from the roof. Top row images are from 7x binoculars and bottom row from 10x binocular system. See supplementary video, available online.

observation roof as two different binoculars with 7x and 10x magnification factors are scanning the area. The geo-located markers have been directly selected in Google Earth, and the tags appear at the correct spots when the buildings are in the binoculars' view. Also, tags do not jump from one object to another and not jitter or drift as binoculars pan around.

A more complete information sharing example is depicted in Fig. 11. There, places of interests are labeled via the Google Earth applet that runs in the browser. Furthermore, since the absolute orientation of the device is tracked, in addition to simple target augmentation we introduce navigation arrow that is meant to help to find labels of interest. This functionality is proved to be particularly useful in devices with large augmentation and narrow field of view, like binoculars, because peripheral vision is not available to the observer the device must be pointed directly to the target in order to see it.

In general, the orientation estimation accuracy is crucial for the performance of the proposed augmented reality binoculars. To evaluate it, the system is operated on the roof of our building looking toward the university campus. The top center of three tallest buildings are augmented with markers (as geo-located Google Earth markers) and the same mid-roof points are hand labeled in every image of the sequence to gather the ground truth for insertion. Using 10x binoculars, we collected two different video sequences of user panning around to look at the towers. The first sequence, referred to as **Seq 1**, is a typical sequence of operator looking through the hand-held binoculars, while the second sequence, referred to as **Seq 2**, is more challenging as binoculars are looking down and shaken more significantly, which is meant to stress the robustness of the system and simulate the use of the device in active harsh environments. Example snapshots from **Seq 1** is depicted in the bottom row of Fig. 10, subsets of both video sequences are given in supplemental material, available online, while Fig. 12 displays the corresponding error statistics in terms of image pixel insertion error as well as angular error of the estimated orientation. As expected, the more challenging scenario **Seq 2** yields a higher root mean squared (RMS) error rate of $RMS_{pixel} = 2.57$ and $RMS_{angle} = 0.0190^{\circ}$ than **Seq 1** which exhibits $RMS_{pixel} = 1.88$ and $RMS_{angle} = 0.0139^{\circ}$ errors. In Fig. 12, abscissa marks the occurrence of each label and do not consider frames where augmented buildings are not visible. Noticeable spikes in insertion error occur once the augmented building is spotted after some time of not actively observing the scene (for instance binocs are looking down, etc) and can be attributed largely due to drift accumulation during dead reckoning portion when global measurements and even frame-to-frame visual feature tracking are unavailable. In conclusion, our system consistently yields very low errors within the accuracy of the hand labeling and markers are correctly inserted both when

Fig. 11. Augmented reality example with geo-landmarks dynamically inserted via Google Earth Applet in the browser. This example shows three specified markers "Tower 1", "Tower 2" and "Sniper" as well as manual mark specified by the binocular operator "manual D". Directional arrow in the upper left corner assists operator in finding specific label in the very narrow field of view 10x binocular image. Switching between existing labels and manual marking of new labels are done via two buttons on the binoculars.



Seq 1 dataset; $RMS_{pixel} = 1.88$, $RMS_{angle} = 0.0139°$ **Seq 2** dataset; $RMS_{pixel} = 2.57$, $RMS_{angle} = 0.0190°$

Fig. 12. Quantitative evaluation of the pose estimation accuracy for datasets **Seq 1** (left) and **Seq 2** (right). Plots show the pixel and equivalent angular error for every hand-labeled marker occurrence. Note that the ground truth point numbers shown in the x-axes are not necessarily continuous in time.

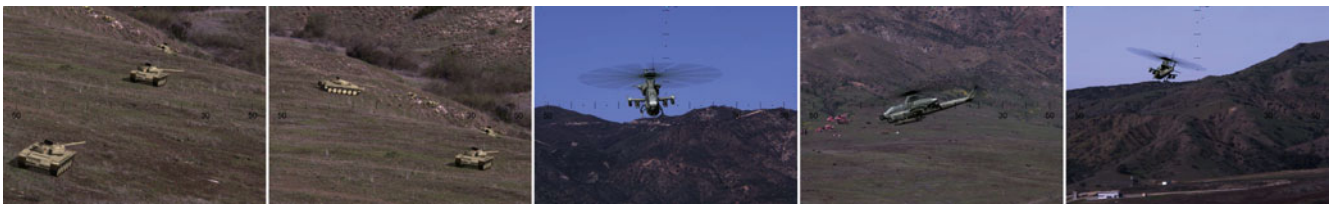


Fig. 13. Augmented reality example for a hand-held binoculars. Augmentation with high quality renderings of tanks and helicopters.

Fig. 14. Augmented reality example for a hand-held binoculars. Images are augmented by stationary truck and flying helicopter. See supplementary video, available online.

Fig. 15. Augmented reality example for an unaided-eye and binoculars setup. Truck and explosions are inserted into unaided-eye and binoculars images. Top row: augmented image from unaided-eye. Bottom row: augmented image from binoculars. See supplementary video, available online.

location for augmentation goes outside the field of view for prolonged times and in the presence of significant vibrations.

3.2 Training Application Example

We show a number of examples from our current AR system developed for training forward observers directing the artillery and aircraft fire. Example augmentations are depicted in Fig. 13 to demonstrate the quality and realism of the rendering. The actual training system consists of the binoculars and the "unaided-eye" system on a helmet. The unaided-eye system is conceptually identical to the binoculars, but has only the wide FoV color camera that is employed for tracking and augmentation – the actual EKF-based 6-DoF pose estimation procedure is described in [18] from which the present algorithm stems.

We present augmentation examples of system being exercised at two independent locations—the roof of our building on the US East Coast and the mountainous terrain in the US West Coast.

For the East Coast location, Fig. 14 shows the field with inserted truck and flying helicopter when tracking them with 7x binoculars. Fig. 15 shows a truck on the field with numerous explosions and smoke effects. Views from both the unaided-eye and binoculars are shown: note that even for such not far distances it is hard to spot the vehicle in the unaided-eye system and binoculars are necessary in this case. Ability to observe augmented reality scene through the combination of the unaided-eye and binoculars gives a very compelling feeling of immersion and is extremely useful for training applications that require high level of realism.

Fig. 16. Augmented reality example for a 7x binoculars setup. Helicopter and tanks under attack are inserted into binoculars' view. See supplementary video, available online.

Fig. 17. AR example for an unaided-eye and two binoculars setup. Tanks and explosions are inserted into helmet and binoculars images. Left: augmented image from unaided-eye. Middle: augmented image from 7x binoculars. Right: augmented image from 10x binoculars. Tank is marked in all three views. See supplementary video, available online.

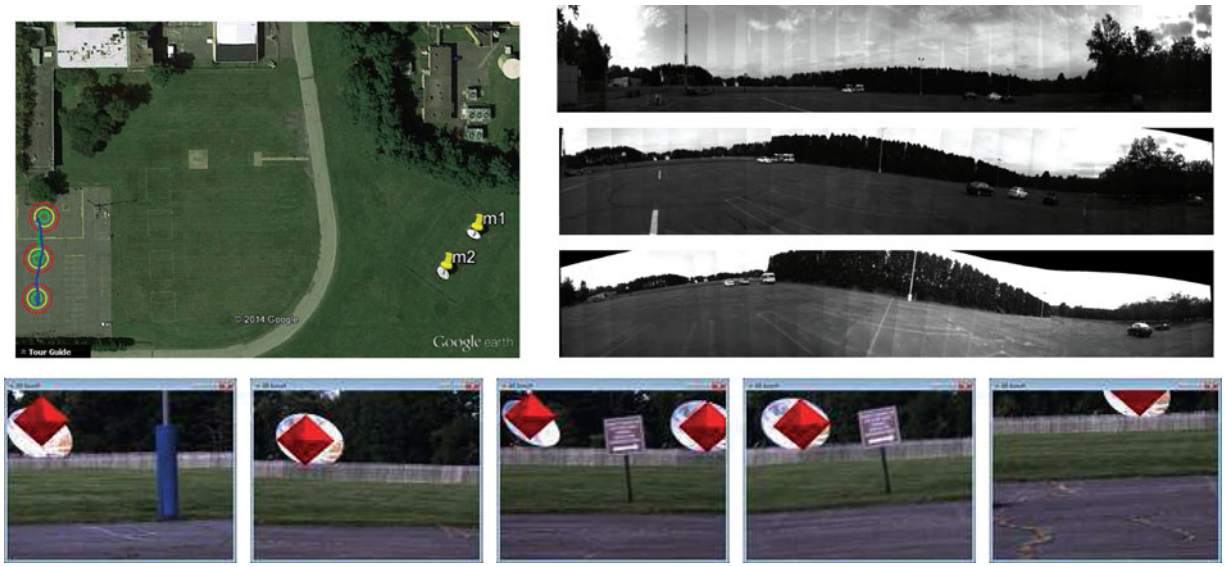


Fig. 18. Augmented reality example with geo-landmarks corresponding to two satellite dishes. Top left: Google earth view of the scene with satellite dishes marked with pins and three panorama locations marked with concentric circles. Top right: Stitched panorama images taken from three locations. Bottom row: Augmented screen shots of the binocs video from different locations. Full video is in supplementary material, available online.

For the West Coast, Fig. 16 depicts a helicopter flying around and attacking a tank located next to another burning tank. The action happens at a distance of about 1 kilometer and binoculars are required to see it in details. Figure 17 shows artillery fire scenario of mortars hitting the tanks. The field is simultaneously observed with unaided-eye, 7x and 10x binoculars. Importantly, the same tank is visualized in all three devices (marked with yellow ovals in Fig. 17) to demonstrate that insertions are performed at the same location relative to the scene in all AR devices.

3.3 Mobile AR-Binoculars

The first experiment of mobile scenario is on a dataset from a parking lot in which the binoculars are continuously moving from near the top of the parking lot to the bottom as shown on the left hand side of the Fig. 18, while trying to maintain the two satellite dishes in the field of view at all times (marked by the two pins on the right hand side). In the figure, the trajectory output of the filter is shown in green and the output from GPS is shown in blue. In addition, the three concentric circles (diameters of 5, 10 and 15

Fig. 19. Augmented reality example with vehicles. From left to right: moving trajectories overlaid on Google Earth view with 3D models; Insertions from two different locations on the balcony. Both augmented narrow FoV and the raw wide FoV images are shown to better visualize the change in viewpoint. Full video is in supplementary material, available online.

meters each) are shown to depict the spots where we have collected our panoramas, which are displayed in a stitched form. Finally, screen shots of red diamonds with geodetic locations corresponding to the satellite dishes on Google Earth are viewed from various locations of the trajectory.

In comparison to the first example which demonstrates ground level mobility, in the next example we present results from an elevated viewpoint (two corner locations of a building balcony), which constitutes to a more realistic use case of binoculars. Instead of constant movement as in the first example, the second dataset involves transitionary movement between the two balcony corners, as the user spends more time in observing from these two spots while occasionally changing location between the two spots (for which we have created two panoramas beforehand). Fig. 19 shows two screen shots of a virtual tank as viewed by the binoculars from different locations on the balcony. The trajectory output of the filter is shown in green and the output from GPS is shown in blue. The supplementary video, available online, shows multiple virtual tanks scattered throughout the campus.

In order to fully demonstrate drift and jitter free experience offered by the proposed augmented reality binoculars, these and additional video examples with audio are provided in the supplementary material, available online, accompanying this paper.

4 Conclusion

In this paper we presented a concept of Augmented Reality Binoculars. We have developed a robust and accurate error-state EKF-based algorithm to track the 6-DoF pose using IMU, GPS, wide FoV and narrow FoV cameras. Furthermore, we added visual landmark matching and panorama mechanism to allow for rapid global correction of orientation and minimize the drift in global orientation estimation. The AR binoculars concept was realized in hardware and operates with low latency providing accurate jitter-free augmentation of the real scene. We believe that AR binoculars has a great potential to be utilized in numerous applications that include but not limited to training and information sharing.

The future work will concentrate on implementing the pose tracking and rendering modules on a mobile computational board (e.g., cell phone form factor) to include computation hardware inside the binocular shell and make the device completely portable and self-sufficient. Another important ongoing work diretion is to support even greater range of free-form movement by the user. In order to address these issues we are currently implementing the framework to use the unaided-eye and binocular platforms jointly by sharing information across the two systems. In particular, the motion estimation pipeline running on the unaided-eye system acts as the master and continuously transmits the head position and image data to the binocular system. As a result, while the user is on the move with the binoculars pointing down, the sensor fusion algorithm running on the binocular system will be using these position estimates as additional sensor updates on its own position (in place of its GPS readings, which are of inferior quality and less frequent). In addition, a continuous local panorama based on the images sent from the helmet camera is being built every few meters covering a time duration of the past several minutes to allow for rapid heading initialization of the binoculars as soon as they are raised to the eye level. We expect that combining these additional capabilities will further enhance the mobility of our immersive training system.

Acknowledgments

This work was supported by ONR Project: Augmented Immersive Team Training (AITT) under Contract N00014-11-C-0433. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of ONR. The authors would like to thank Richard Schaffer and Sean Cullen of Lockheed Martin, Burlington, Massachusetts, for the simulation engine used in rendering the synthetic elements for some of the experimental results.

References

[1] C. Arth, M. Klopschitz, G. Reitmayr, and D. Schmalstieg, "Real-time self-localization from panoramic images on mobile devices," in *Proc. 10th IEEE Int. Symp. Mixed Augmented Reality*, 2011, pp. 37–46.

[2] R. Azuma, B. Hoff, H. Neely, and R. Sarfaty, "A motion-stabilized outdoor augmented reality system," in *Proc. IEEE Virtual Reality*, 1999, pp. 252–259.

[3] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *Int. J. Comput. Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[4] K. W. Chia, A. D. Cheok, and S. Prince, "Online 6 DOF augmented reality registration from natural features," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2002, pp. 305–313.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recog.*, 2005, pp. 886–893.

[6] M. Fiala, "ARTag, a fiducial marker system using digital techniques," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recog.*, 2005, vol. 2, pp. 590–596.

[7] S. Gauglitz, C. Sweeney, J. Ventura, M. Turk, and T. Hollerer, "Live tracking and mapping from both general and rotation-only camera motion," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2012, pp. 13–22.

[8] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vision Conf.*, 1988, pp. 147–152.

[9] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge Univ. Press, Cambridge, U.K., 2004.

[10] B. Jiang, U. Neumann, and S. You, "A robust hybrid tracking system for outdoor augmented reality," in *Proc. IEEE Virtual Reality*, 2004, pp. 3–275.

[11] K. Satoh, M. Anabuki, H. Yamamoto, and H. Tamura, "A hybrid registration method for outdoor augmented reality," in *Proc. IEEE/ACM Int. Symp. Augmented Reality*, 2001, pp. 67–76.
[12] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. 6th IEEE/ACM Int. Symp. Mixed Augmented Reality*, 2007, pp. 225–234.
[13] J. B. Kuipers, *Quaternions and Rotation Sequences*. Princeton, NJ, USA: Princeton Univ. Press, 1998.
[14] D. Kurz and S. Benhimane, "Gravity-aware handheld augmented reality," in *Proc. 10th IEEE Int. Symp. Mixed Augmented Reality*, 2011, pp. 111–120.
[15] F. M. Mirzaei and S. I. Roumeliotis, "A Kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1143–1156, Oct. 2008.
[16] A. Mourikis and S. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 3565–3572.
[17] R. A. Newcombe, S. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Proc. IEEE Int. Conf. Comput. Vision*, 2011, pp. 2320–2327.
[18] T. Oskiper, S. Samarasekera, and R. Kumar, "Multi-sensor navigation algorithm using monocular camera, IMU and GPS for large scale augmented reality," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2012, pp. 71–80.
[19] S. Prince, K. Xu, and A. D. Cheok, "Augmented reality camera tracking with homographies," *IEEE Comput. Graph. Appl.*, vol. 22, no. 6, pp. 39–45, Nov./Dec. 2002.
[20] D. Pustka, J.-P. Hulss, J. Willneff, F. Pankratz, M. Huber, and G. Klinker, "Optical outside-in tracking using unmodified mobile phones," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2012, pp. 81–89.
[21] G. Reitmayr and T. Drummond, "Going out: Robust model-based tracking for outdoor augmented reality," in *Proc. IEEE/ACM Int. Symp. Mixed Augmented Reality*, 2006, pp. 109–118.
[22] S. Roumeliotis and J. Burdick, "Stochastic cloning: A generalized framework for processing relative state measurements," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2002, pp. 1788–1795.
[23] G. Simon, "Tracking-by-synthesis using point features and pyramidal blurring," in *Proc. 10th IEEE Int. Symp. Mixed Augmented Reality*, 2011, pp. 85–92.
[24] C. A. L. Waechter, M. Huber, P. Keitler, M. Schlegel, G. Klinker, and D. Pustka, "A multi-sensor platform for wide-area tracking," in *Proc. 9th IEEE Int. Symp. Mixed Augmented Reality*, 2010, pp. 275–276.

Taragay Oskiper received the BSc (Honours) degree in electrical engineering from Bilkent University, Ankara, Turkey, in 1996, and the MSc and PhD degrees in electrical engineering from Princeton University, New Jersey, in 1998 and 2001. From 2001 till 2005, he worked in a biomedical startup company developing signal processing algorithms and time delay neural network methods for detection and classification of heart murmurs in digital auscultation. Since 2005, he has been with SRI International, Princeton, New Jersey, where he is currently a senior principal research scientist. His areas of expertise are in the broad field of vision aided navigation algorithms as related to augmented reality, motion estimation and geo-registration for ground and aerial platforms, multisensor fusion, and in particular visual inertial odometry.

Mikhail Sizintsev received the PhD degree in computer science from York University, Toronto, Canada, in 2012. He spent the summer 2009 at Sarnoff Corporation in Princeton, New Jersey, as an intern developing GPU-based stereo systems for augmented reality applications. He is currently a computer scientist at SRI International in Princeton. His major areas of research include stereo, motion, augmented reality, and multisensory navigation. He received the Canadian Image Processing and Pattern Recognition Society (CIPPRS) Doctoral Dissertation Award for his thesis "On 3D Spacetime Oriented Energy Representation for Spatiotemporal Stereo and Motion Recovery." He is a member of the IEEE.

Vlad Branzoi received the BS and MS degrees in computer engineering and computer science from Columbia University, New York City, in 2001 and 2003, respectively. He is currently a senior computer scientist at SRI International, Princeton, New Jersey. He has more than 10 years experience in building novel sensors and integrated multisensor systems for training, robotics, and mobile applications. He is a member of the IEEE.

Supun Samarasekera received the MS degree from the University of Pennsylvania, Philadelphia. He is currently the technical director of the Vision and Robotics lab. at SRI International, Princeton, New Jersey. Prior to joining SRI, he was employed at Siemens Corp. He has more than 15 years of experience in building integrated multisensor systems for training, security, and other applications. He has led programs for robotics, 3D modeling, training, visualization, aerial video surveillance, multisensor tracking, and medical image processing applications. He has received number technical achievement awards for his technical work at SRI. He is a member of the IEEE.

Rakesh Kumar received the BTech degree in electrical engineering from the Indian Institute of Technology, Kanpur, India, in 1983, the MS degree in electrical and computer engineering from the State University of New York at Buffalo in 1995, and the PhD degree in computer science from the University of Massachusetts at Amherst in 1992. He is the director of the Center for Vision Technologies in Information and Computing Sciences at SRI International. In this role, he is responsible for leading research and development in the fields of computer vision, robotics, image processing, computer graphics, and visualization algorithms and systems for government and commercial clients. The Center for Vision Technologies has about 100 researchers and is one of premier industrial research labs in computer vision in the world. In 2013, he was honored with the Outstanding Achievement in Technology Development award from his alma mater, University of Massachusetts Amherst, School of Computer Science. He has received the Sarnoff Presidents Award in 2009 and Sarnoff Technical Achievement awards for his work in registration of multisensor, multidimensional medical images and alignment of video to three-dimensional scene models. He has also been an associate editor for the *IEEE Transactions on Pattern Analysis and Machine Intelligence*. He has co-authored more than 60 research publications, and received more than 50 patents. He was a principal founder for multiple spin-off companies from Sarnoff Corporation, including VideoBrush, LifeClips, and SSG. He is a member of the IEEE.