

# A Robust Hybrid Tracking System for Outdoor Augmented Reality

Bolan Jiang    Ulrich Neumann    Suyu You

*Integrated Media Systems Center  
University of Southern California*

{boljiang | uneumann | suyay}@graphics.usc.edu

## Abstract

*We present a real-time hybrid tracking system that integrates gyroscopes and line-based vision tracking. Gyroscope measurements are used to predict pose orientation and image line feature correspondences. Gyroscope drift is corrected by vision tracking. System robustness is achieved by using a heuristic control system to evaluate measurement quality and select measurements accordingly. Experiments show that the system achieves robust, accurate, and real-time performance for outdoor augmented reality.*

## 1. Introduction

An augmented reality (AR) application can enhance a user's perception by fusing computer-generated virtual information into their view of the real world. AR has many potential applications in military, manufacturing, entertainment, and medical domains. Ideally, the virtual information should maintain a correct spatial relationship with the real world. A key capability for accomplishing correct fusion is a tracking system that accurately estimates the user's 6DOF pose (or camera pose, if the user views the world through a camera).

A wide variety of tracking technologies have been developed for AR [1,14]. Among all the tracking approaches, model-based vision is often used to achieve high accuracy [14]. A critical process in these systems matches (or corresponds) image features with features of a 3D scene model. This feature correspondence process is often computationally intensive and prone to failure. Some systems address correspondences by using specially designed markers. However, this is not practical in outdoors environments. Temporal coherence is often relied upon to obtain real-time feature correspondences for natural features [4,5,22]. Feature search is thereby limited to small displacements around predicted positions. Large feature displacements, arising from rapid motions or long intervals, often exceed the search space and result in failures to compute correspondences.

Other sensors [2,3,18] have been integrated with vision to help in predicting image feature movements, and these

hybrid systems can perform more robustly with large image feature displacements.

This paper presents a real-time tracking system for outdoor augmented reality in urban settings where building models are available. The system uses gyroscopes (gyros) and vision tracking of line-features visible in the scene. Gyro and vision measurements are fused by a heuristic assessment process and an extended Kalman filter. Gyroscope measurements are used to predict pose orientation and line feature correspondences. Gyro drift is stabilized by vision tracking. Real-time robust performance is obtained as evident from the results we present. The system successfully tracks over long periods of time, with high camera motion rates, with temporary loss of visible features, and with dynamic occlusions.

Section 2 reviews related work; section 3 details the tracking system; and section 4 presents the experiments and analyses their results.

## 2. Related Work

Global Positioning Satellites (GPS), electronic compasses, and inertial sensors are widely used in outdoor tracking systems. For example, differential GPS and gyros are used in [15]. GPS and a compass are used in [16]. Compared to model-based vision, these tracking technologies are often more robust but less accurate [14].

Due to the high accuracy of model-based vision methods, many standalone and hybrid vision systems have evolved. Indoor systems [8,9] often use artificial markers to simplify correspondence computations. In [6,11,13], magnetic sensors, accelerometers, and gyroscopes are combined with marker-based vision methods. The added sensors predict feature positions to speed up feature detection and correspondence calculations. Our system uses gyro sensors for similar purposes, however, it detects and corresponds line features on buildings that occur in outdoor environments and it selectively processes vision data to increase robustness under non-ideal imaging conditions.

Vision systems can track natural features [4,5,22] by using temporal coherence to aid in real-time feature correspondence. However, these systems often fail when

presented with momentary occlusions or large image displacements from rapid motion. In [19], inter-frame affine transformations are recovered by tracking planar regions. Planar regions can be corresponded with large image displacements. In the absence of a 3D model, only 2D texture can be augmented in the real world.

Inertial sensors have been fused with vision tracking system to achieve robustness for rapid motion. Several hybrid systems use point-features and inertial sensors for pose recovery. In [2], gyros are used to predict the positions of 2D image points, and the orientation recovered by points is used to correct the drift of gyros. In [3], the accelerometer and gyros are used to predict 6DOF pose for the vision system. Our system tracks lines rather than points to recover pose. Lines are prominent in outdoor urban settings since most buildings provide numerous line features. Lines often remain in the view much longer than points, since lines can extend over an entire structure; lines are useful, even when partially occluded; and lines can be localized accurately since there are often many supporting pixels along a line.

A theoretical proof is provided in [17] that 2D line motion tracking can correct the drift of gyroscopes, however, there is no experimental implementation provided. Klein et al [18] integrate line features and gyroscopes for an indoor tracking system. Our system is designed for building line features, and validated in outdoor environments.

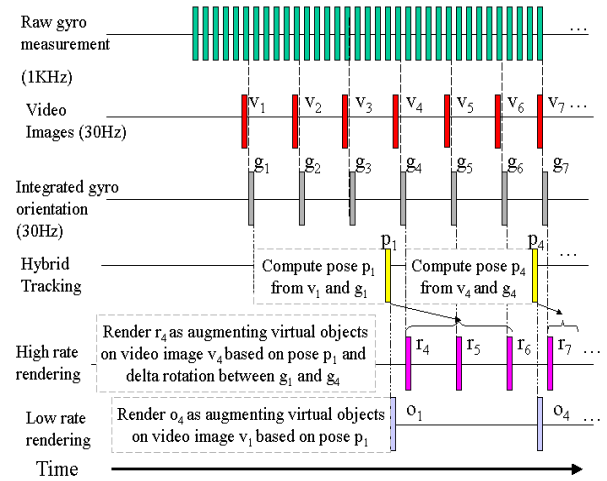
Another vision approach uses similarity to key frames to recover camera pose [20, 21]. In an offline process, key frames are captured and their camera pose estimated. In an online process, the current camera pose is recovered by registering the current image to stored key frames. Scene changes, occlusions, and varying imaging conditions limit the robustness of these approaches.

### 3. Hybrid Tracking Method

Our tracking system requires an initialization stage where the user needs to manually correspond some line features of the 3D model with detected 2D lines. The pose for the captured image is estimated from these manually matched pairs. While it is possible that this initialization process could be simplified or automated by using GPS and compass data, our current approach is manual since we only use gyro and vision sensors.

Once an initial pose is computed, the system switches to tracking mode. The initial image and corresponded lines are stored as reference information for later line detection.

As shown in Figure 1, the raw gyro measurements are captured at 1KHz and video images are captured around 30Hz in the tracking stage. After capturing an image, the absolute orientation and current rotation rate are computed from the gyro measurements and the prior (or initial) pose estimate. The orientation and rotation rate are used with the captured image to estimate a new pose. There are two



**Figure 1** - Temporal relationships of capturing, tracking, and rendering. Low-rate rendering only renders the images processed by vision tracking. High-rate rendering augments each image based on the latest pose estimate and the orientation changes measured by gyros. Low rate rendering is used for the experiments shown in this paper.

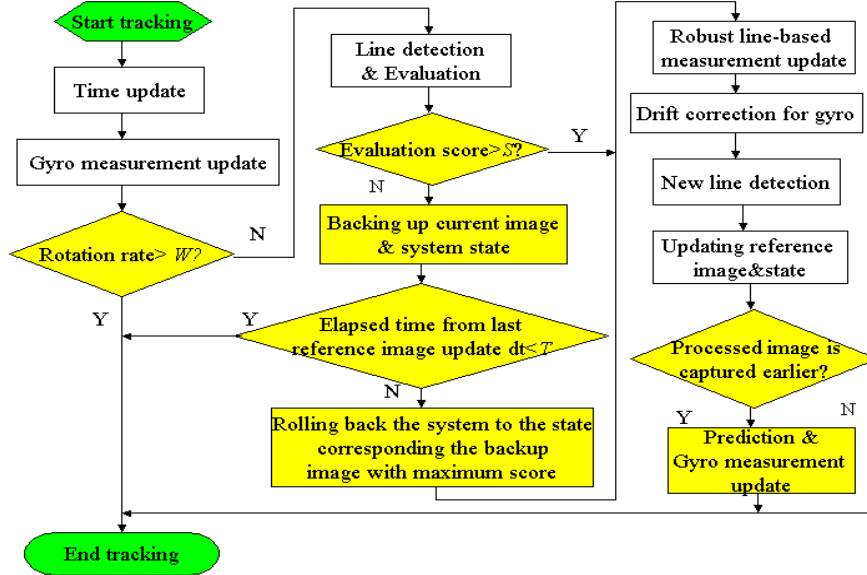
methods for rendering augmented images. A low-rate rendering method only renders the processed images at whatever frame rate is computationally feasible (3-8 Hz typical in our system). Low rate rendering ensures accurate registration but is only available at the vision processing rate. We use the low rate rendering method in all of the tracking experiments to visualize the tracking system performance.

A high rate rendering method could use the latest pose estimate and changes in orientation measured by the gyro to render each captured image. Figure 1 shows how virtual objects are augmented on image  $v_4$ , for example, using the latest pose estimate ( $p_1$ ) and the rotations measured by the gyro between  $g_1$  and  $g_4$ . High rate rendering decouples the rendering rate from the vision tracking process rate to achieve higher update rates. Although registration errors can accumulate between vision pose updates, these are typically small as the gyros are very accurate over short periods of time. Similar ideas of asynchronous measurements and pose updates are described in [8].

An Extended Kalman filter (EKF) is used to integrate measurements from gyroscope and visual sensors. As in [9], the EKF state vector is defined as  $\vec{e} = (x, y, z, \dot{x}, \dot{y}, \dot{z}, \Delta\theta, \Delta\phi, \Delta\varphi, \dot{\theta}, \dot{\phi}, \dot{\varphi})$ , where  $(x, y, z)$  represent position and  $(\Delta\theta, \Delta\phi, \Delta\varphi)$  are Euler angles representing the incremental rotation. The global rotation is stored in an external quaternion  $q = (q_w, q_x, q_y, q_z)$ .

#### 3.1. Robust Hybrid Tracking

The components of the tracking system are illustrated in Figure 2. The white blocks are operations related to



**Figure 2** - The structure of the hybrid tracking system. The white blocks represent pose estimation operations, and the yellow blocks represent the heuristic control operations.

pose estimation, while the yellow blocks relate to the heuristic control system. The heuristic control system only chooses the high quality visual measurement for vision tracking.

When gyro and vision measurements are available, the current pose is predicted by time update. Orientation prediction is then updated by gyro measurement update. More details about time update can be found in [9].

Hybrid tracking systems [2,3,18] require synchronization among different sensors, which is difficult to achieve in practice. Good performance can be obtained with approximate synchronization if the vision processes are skipped during high motion rates. Unless the gyroscope and visual sensor are accurately synchronized, rapid camera rotations will cause large errors in the image feature positions predicted by the gyros. Large prediction errors cause 2D line detection and correspondence calculations to fail. In our system, when the rotation rate measured by the gyro exceeds a threshold  $W$  ( $\sim 20$  degree/sec), further vision operations are skipped and the algorithm starts a new time update.

For rotation rates below the threshold, vision processing starts with 2D line detection. The detection algorithm searches for line segments in the neighborhood predicted by gyro measurements from the time of the reference image. (The reference image is the last image in which lines were detected and corresponded and pose was computed by vision tracking operations.) The quality of the detected lines is evaluated by computing a similarity between the predicted and detected lines (as described in Section 3.3). If the evaluation score exceeds a threshold  $S$ , the detected and corresponded line pairs are used to update pose, which in turn is used to correct gyro drift and detect new lines. A successfully processed image is saved as the new reference image. If the evaluation score is lower than

the threshold and the elapsed time from last reference image update is shorter than a time threshold  $T$  ( $\sim 1.5$  seconds), the remaining vision operations are suspended and the current image, feature data and state for the tracking system are saved. The algorithm starts the next time update and a new image is examined.

If conditions are such that no images pass the threshold tests of evaluation  $S$  for a period exceeding  $T$ , the system uses the stored image and feature data with the best evaluation score. If more than one image have the same score, the latest is selected. The tracking system is restored to the state corresponding the selected image. The pose for the selected image is estimated by vision operations. If the selected image is captured earlier, the pose for current image is predicted from this selected image and updated by gyro measurement.

### 3.2. Gyro Update and Drift Correction

The absolute orientation measured by the gyros in the gyro-body coordinate frame is represented by the rotation matrix  $R_g$ . The rotation matrix  $R_{wc}$  for the camera in world coordinate frame can be inferred as

$$R_{wc} = R_{wg} R_g R_{gc},$$

where  $R_{wg}$  is the rotation between the world and gyro-body coordinate frames and  $R_{gc}$  is the rotation between gyros and camera local coordinate frames. Both  $R_{wg}$  and  $R_{gc}$  are constants calibrated before tracking. Then the measurement vector for gyroscope is computed as  $z_g = (\Delta\theta_g, \Delta\phi_g, \Delta\varphi_g) = EulerAngleFrom(R_q^{-1} \Delta R_d R_{wc})$  where  $R_q$  is the rotation matrix computed from the

quaternion representing the current global orientation estimate.  $\Delta R_d$  is the drift correction matrix and is initialized to identity matrix in initialization stage. And the predicted pose estimate  $\hat{e}^-$  and variance  $\hat{P}^-$  for the pose are updated as follows:

$$\begin{aligned} K &= \hat{P}^- H_g^T (H_g \hat{P}^- H_g^T + \Xi)^{-1} \\ \hat{e}^g &= \hat{e}^- + K(z_g - H_g \hat{e}^-) \\ \hat{P}^g &= (I - KH_g) \hat{P}^- \end{aligned} \quad (1)$$

where  $H_g = [0_{3 \times 3} \quad 0_{3 \times 3} \quad I_{3 \times 3} \quad 0]$ , and  $\hat{e}^g, \hat{P}^g$  are the updated pose and variance estimate.  $\Xi$  is a 3x3 diagonal noise matrix for gyro measurements.

If the pose is updated by vision tracking, the correction matrix is updated as

$$\Delta R_d = \hat{R} R_{wc}^{-1}$$

where  $\hat{R}$  is the orientation estimate from vision measurement update.

### 3.3. 2D Line Detection and Evaluation

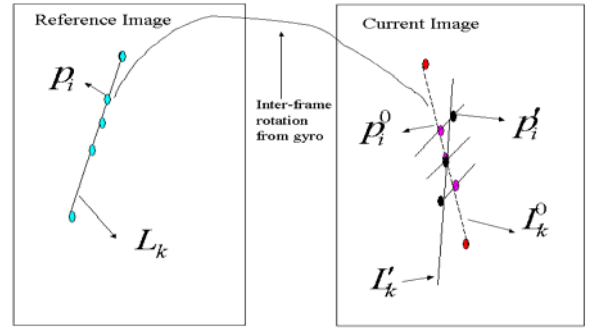
The reference image contains lines that were detected and corresponded with line features in a 3D scene model. If the motion between the current and reference images is purely rotation  $\Delta R$ , the point  $p_i = (x_i, y_i)$  on the reference image matches the point  $p_i^0 = (x_i^0, y_i^0)$  on the current image as follows:

$$\begin{cases} x_i^0 = \frac{d_{11}x_i + d_{12}y_i + d_{13}}{d_{31}x_i + d_{32}y_i + d_{33}}, \\ y_i^0 = \frac{d_{21}x_i + d_{22}y_i + d_{23}}{d_{31}x_i + d_{32}y_i + d_{33}} \end{cases} \quad (2)$$

where  $D = \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{bmatrix} = K \Delta R K^{-1}$  and  $K$  is the

intrinsic matrix for the camera. When the inter-frame translation is non-zero but small compared with the viewing distance,  $p_i^0$  is still a good prediction for the correspondent to  $p_i$ . In outdoor environments where the model is relatively far from the user, the image feature displacements caused by slow speed (walking) translation over a short time can be ignored compared to the displacements caused by rotations.

As shown in Figure 3, any point (cyan points) on a 2D line  $L_k$  on the reference image is predicted on the current image. The two predicted end points (red points) form a predicted line  $L_k^0$ . For each point  $p_i$  on the reference image, search its correspondent  $p_i'$  in a 1D interval  $\{q_j' \in [-J, J]\}$  centered at the predicted point  $p_i^0$  in the direction normal to the predicted line as follows:



**Figure 3** 2D Line detection. Line correspondences are searched around the lines predicted by gyro measurements.

$$p_i' = \arg \max_{q_j' \in [-J, J]} |G_j'| \quad (3)$$

subject to  $\alpha < \frac{G_i'}{G_i} < \beta$  and  $|G_i| > \eta$ .

$G_j'$  is the gradient in the direction normal to the predicted line for point  $q_j'$  on the current image.  $G_i$  is the gradient for  $p_i$  in the direction normal the line on the reference image. The directional gradient is computed by the convolution masks defined in [7].  $\alpha$  is a constant smaller than 1 while  $\beta$  is a constant greater than 1.  $\alpha$  and  $\beta$  are used to ensure the similarity between the two gradients. In our implementation,  $\alpha = 0.5$  and  $\beta = 5$ .  $\eta$  is the gradient threshold used to choose the pixels with a significant intensity discontinuity. The search distance  $J$  is chosen as 3 pixels in our implementation. The detected points are fit into a straight line  $L_k'$  by a Hough transform.

To evaluate the quality of the detected lines, a similarity score is computed as follows:

$$s = \frac{1}{N} \sum_{i=1}^N \frac{l_k'}{l_k^0}, \quad (4)$$

where  $N$  is the number of predicted lines,  $l_k'$  is the length for the  $k^{\text{th}}$  detected line and  $l_k^0$  is the length for the corresponding predicted line. Ideally, if the gyro measurements are accurate and the inter-frame translation is small, the predicted lines should be very close to the correct positions. So the detected line should have a length similar to the predicted length, resulting in a value of  $s$  close to one. On the other hand, a small  $s$  means the predicted lines may be inaccurate. Poor predictions increase the probability that correspondences are lost and the vision system is detecting and tracking the wrong lines.

### 3.4. Robust Line Measurement Update

Line measurement update uses corresponded line pairs to update the current pose. In our system, a 3D model line

is represented by its two end points  $V_{i1}$  and  $V_{i2}$ . A 2D detected line is represented by its two end points  $(x_{i1}, y_{i1}), (x_{i2}, y_{i2})$  on the image. A 3D model line is projected as an infinite 2D line on the image as follows:

$$\begin{bmatrix} m_{ix} \\ m_{iy} \\ m_{iz} \end{bmatrix} = K^{-T} R^{-1} \left( (V_{i1} - \begin{bmatrix} x \\ y \\ z \end{bmatrix}) \times (V_{i2} - \begin{bmatrix} x \\ y \\ z \end{bmatrix}) \right) \quad (5)$$

where  $K$  is the intrinsic matrix for the camera,  $(x, y, z)$  is the camera position and  $R$  is the rotation matrix computed from the incremental rotation  $(\Delta\theta, \Delta\phi, \Delta\varphi)$  and global orientation  $q$ . The projection error is defined as the squared distances from the detected end points to the projected line as follows:

$$E_i^2 = f(\bar{e}, x_{i1}, y_{i1}, x_{i2}, y_{i2}) = d_{i1}^2 + d_{i2}^2, \quad (6)$$

$$d_{i1}^2 = \frac{(x_{i1}m_{ix} + y_{i1}m_{iy} + m_{iz})^2}{m_{ix}^2 + m_{iy}^2}$$

where

$$d_{i2}^2 = \frac{(x_{i2}m_{ix} + y_{i2}m_{iy} + m_{iz})^2}{m_{ix}^2 + m_{iy}^2}$$

Though line feature similarity is evaluated and images with low similarity score are skipped, it is still possible to have outliers in the matched pairs. The outliers are first detected by a Least Median Square [12] method. Groups of three line pairs  $(m, n, k)$  are randomly drawn from the  $N$  matched line pairs. The chosen lines are used to estimate pose by minimizing  $E_m^2 + E_n^2 + E_k^2$ . Then the projection errors for all the pairs are computed. The group with the minimum median error is selected. The robust standard deviation estimate is given by

$$\sigma = 1.4826 [1 + 5/(N - 3)] \sqrt{M_J} \quad (7)$$

where  $M_J = \text{med}_{i=1, \dots, N} r_i^2$ ,  $r_i^2, i = 1, \dots, N$  are the projection

errors corresponding to the selected group. Then the pose is updated by minimizing the following objective function in EKF

$$E = (\bar{e} - \hat{e}^g)^T (\hat{P}^g)^{-1} (\bar{e} - \hat{e}^g) + \sum_{i=1}^N w_i E_i^2, \quad (8)$$

$$\text{where } w_i = \begin{cases} 1 & \text{if } r_i^2 < (2.5\sigma)^2 \\ 0 & \text{otherwise} \end{cases}$$

and  $\hat{e}^g, \hat{P}^g$  are the pose estimate and variance after the gyro measurement update.

The detected lines with zero weight are treated as outliers and deleted.

### 3.5. Detect New Lines

When the user moves around, previously detected lines may disappear and new lines may appear. The newly appeared and lost lines need to be detected to ensure consistent performance.



Figure 4 - Gyroscope and vision tracking sensors

After pose estimation, the 3D model lines are projected on the image based on the current pose estimate. The hidden lines are removed as in [23]. For projected 3D model lines without 2D correspondents, a new line search is performed in their neighborhood. Finding new lines is particularly critical as many factors can cause the loss of currently tracked lines. Unless new lines are constantly sought, the system will eventually lose the set of starting lines, causing pose loss.

For each point on a predicted line, a 1D interval  $[-J, J]$  centered at the point and perpendicular to the projected line is searched. The point with maximum gradient in the direction normal to the predicted line is selected as a line point. All the selected points are fitting into a straight line by Hough transform.

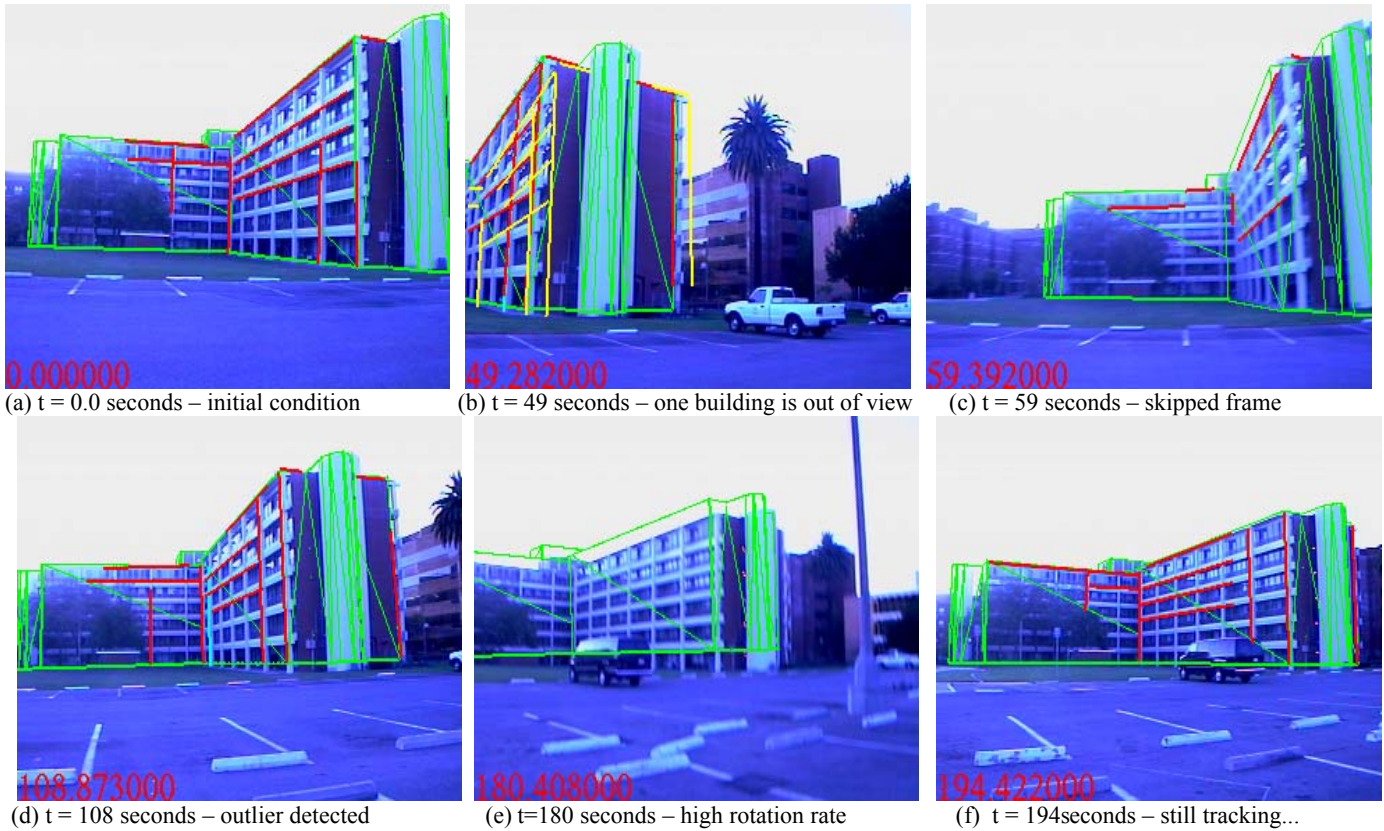
## 6. Experiments

As shown in Figure 4, the hybrid sensor is composed of a CCD video camera (Sony XC-999 with 6mm lens), and three orthogonal rate gyros (GyroChip II QRS14-500-103) made by Systron Donner. The video images are captured by an Osprey frame grabber, and three gyros are sampled via 16-bit A/D converter (National Instrument DAQPCI-AI-16XE-20). The whole capture and tracking system is run on a Dell Pentium-IV 1GHz PC.

In outdoor tests, the user is walking on a parking lot while looking at the buildings besides the parking lot. The wire-frame models of buildings are extracted from LiDAR data as described in [10]. A set of 3D lines is added to the model surfaces to represent the line features of the façade detected during tracking. The addition of these line features is currently done manually based on image textures projected on the wire-frame surfaces. (Clearly, automation of this step is desirable and feasible, but our focus is on the tracking system using the models.)

### 6.1. Off-line Experiments

Initially, we captured images and gyro measurements and processed the tracking off-line. The captured gyro measurements are rotation rate and absolute orientation synchronized to the video capture rate. No hardware synchronization support is used. The program simply reads the video and gyro data in sequence, storing them with time stamps. Note that the gyro data samples are



**Figure 5** - Images from off-line experiments. Green wire-frame model is 3D augmentation. Red lines are the 2D lines detected in camera images. The left-bottom number is the time stamp. (a) Two buildings were in view at the beginning. (b) One building moves out of view during tracking. The yellow lines show lines from prior frame to illustrate image displacements. (c) Vision processing is skipped after low evaluation score for 2D line detection. (d) An outlier (cyan line) is culled due to the close proximity of vertical lines. (e) Vision processing is skipped due to large rotation rate. (f) Continued tracking after 194 seconds.

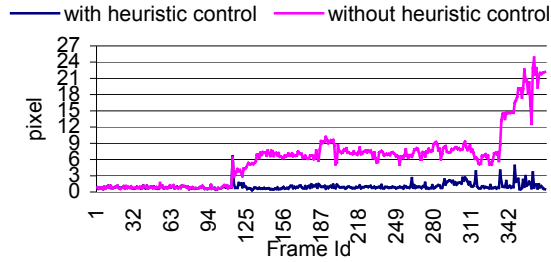
buffered in the A/D converter, so they are accessed in complete blocks.

The user walked from left to right on the parking lot, and then walked backward, and finally walked to right again. The camera was oriented at the buildings during capture. To test the system performance for large image displacements, the images and gyro data were captured at a low rate of 2Hz over the capture period of about three minutes.

Six images from the offline experiment are shown in Figure 5. In each image, the green wire-frame model is the 3D model overlaid on the image based on the pose estimate. The number on the low-left corner is the time stamp (in seconds). Red lines are the 2D lines detected during tracking. At the beginning of the sequence (Figure 5a), two building wings were in view. When the user moved to the right, one building moved out of view (Figure 5b). The yellow lines in Figure 5b show the lines detected in the last frame. By using gyro data for feature prediction, lines can be corresponded correctly, even for the large image displacements shown in Figure 5b. In Figure 5c, the operations after line detection were skipped due to the low evaluation score for the detected lines. In Figure 5d, an outlier (cyan line) appeared due to the close proximity of vertical lines. This outlier was identified in

the robust line measurement update. The correct line (red line left of the cyan line) was detected in the new line detection. In Figure 4e, high rotation rate caused the skipping of vision processing. From the augmented 3D model, it is clear that lines cannot be predicted correctly by gyro measurements during high rate motion due to synchronous errors. The building out of view in Figure 5b is in view again in Figure 5f. All the lost lines were re-detected correctly in Figure 5f. Note the arrival of the van on the scene, partially occluding lines during the sequence.

To evaluate the tracking performance, the 2D lines are detected and manually matched with 3D model lines on each processed image. The average of projection errors defined in equation (6) was computed. The pose was estimated by a tracking system with and without the heuristic control system. Without the heuristic system, the vision tracking is simply processed for each frame. The average errors for the two tracking tests are compared in Figure 6. At the 113<sup>th</sup> frame, which is the snapshot in Figure 5c, the tracking system without the heuristic control processed the low quality visual measurements, causing the pose calculation and future correspondences to diverge. On the other hand, the tracking system with the heuristic control achieved robust performance by skipping the vision operations for low quality images.



**Figure 6** - Average re-projection errors for offline tracking experiments with and without heuristic control

## 6.2. Real-time Experiments

The tracking system was tested outdoor for real-time performance. After an image and gyro measurements were captured, the tracking system estimated pose for the image in real-time, and the 3D model (green wire-frame) was rendered on the captured image with the estimated pose. A DV-camera recorded the images rendered on the computer screen using the graphics card s-video output. The tracking rate is approximately 5Hz. Selected images from the experiment recordings are shown in Figures 7 and 8. Red lines are the detected 2D lines. The number on the left bottom corner is frame number and time stamp separated by a colon.

The first test illustrates the system performance with smooth motions over a long time period. Holding the camera, the user walks slowly from left to right and then back and left. The camera was oriented at the buildings during the four-minute sequence. After initialization, pose is computed automatically for the whole test (Figure 7). In Figure 7c, accurate tracking is maintained as the scene is partially occluded by a foreground pole moving across the image.

The second test stresses system performance for some extreme motions over about 104 seconds. In the example

shown in Figure 8, the camera is quickly rotated away from the modeled buildings and then back. Figure 8a is the image before the rotation. Figure 8b shows the scene during rotation when the modeled buildings are off screen. After the camera rotates back, the pose is recovered accurately, as shown in Figure 8c.

## 7. Conclusions

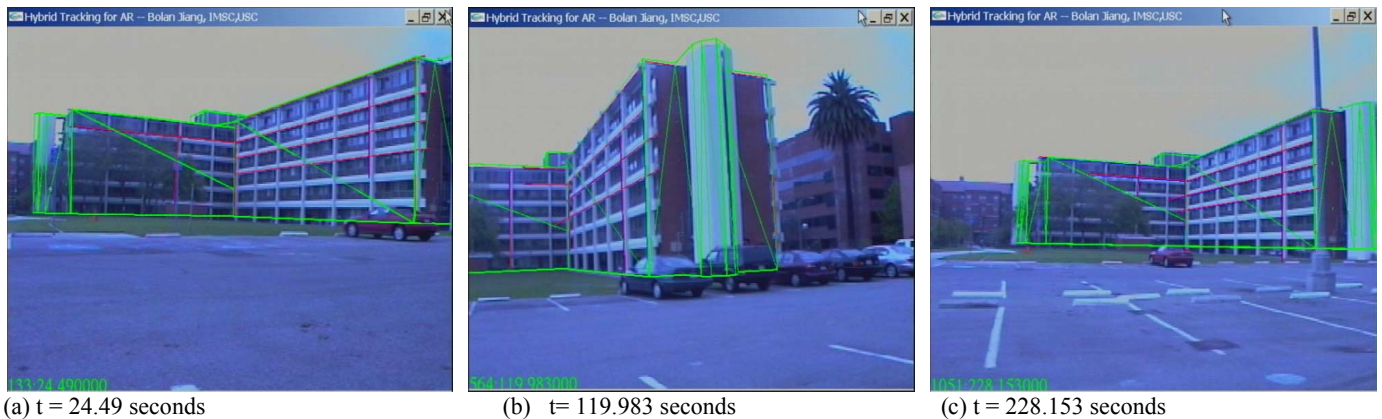
We present a gyros-vision hybrid tracking system for outdoor environments. The vision tracking subsystem uses natural lines for pose recovery. The gyro measurements are used to predict pose orientation and image line features. The novelty of the proposed system is to use a heuristic control system to evaluate measurement quality. And only high quality measurements will be chosen for tracking. The experiment results show that the proposed hybrid system can achieve robust performance in outdoor environments.

## 7. Acknowledgements

This work is supported by Office of Naval Research. Additional funding and research facilities are provided by the National Science Foundation through its funding of the Integrated Media Systems Center.

## 9. References

- [1] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, B. MacIntyre, "Recent Advances in Augmented Reality", IEEE Computer Graphics & Application 21, 6(Nov/Dec 2001).
- [2] K. Satoh, M. Anabuki, H. Yamamoto, H. Tamura, "A Hybrid Registration Method for Outdoor Augmented Reality", Proc. of ISAR'01, pp 67-76.
- [3] M. Ribo, P. Lang, H. Ganster, M. Bradner, G. Stock, and A. Pinz, "Hybrid Tracking for Outdoor Augmented Reality Applications", IEEE Computer Graphics and Applications vol.



**Figure 7** - Images from the first real-time tracking experiment. Green wire-frame model is 3D augmentation. Red lines are the detected lines. Left-bottom corners have frame number: time stamp. Tracking sequence lasts about four minutes. (a) Tracking starts. (b) Middle of sequence. (c) Tracking continues successfully although a near-field pole occludes the building features.



**Figure 8** – Images from the second real-time experiment. The camera rotates away from the buildings. The green wire-frame is a 3D augmentation of the building. Red lines are the detected lines. Left bottom corners show the frame number: time stamp. The whole sequence has multiple rotations in different directions and lasts about 104 seconds. (a) Before rotating away. (b) After the modeled buildings are out of view. (c) Rotation back and automatic recovery of accurate pose.

22. Nov/Dec. 2002, pp54-63.

[4] T. Drummond, R. Cipolla, “Real-time Visual Traking of Complex Structures”, IEEE PAMI Vol. 24, No. 7, July 2002.

[5] E. Marchand, P. Boutheymy, F. Chaumette and V. Moreau, “Robust real-time visual tracking using a 2D-3D model-based approach”, Proc. ICCV’99, pages 262-268.

[6] T. Auer, A. Pinz, “Building a Hybrid System: Integration of Optical and Magnetic Tracking”, IWAR’99, pp.13-22.

[7] P. Boutheymy, “A Maximum Likelihood Framework for Determining Moving Edges”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 5, May 1989.

[8] G. Welch, G. Bishop, “SCAAT: Incremental Tracking with Incomplete Information”, SIGGRAPH 97, pp. 333-344.

[9] J. Park, B. Jiang and U. Neumann, “Vision-based Pose Computation: Robust and Accurate Augmented Reality Tracking”, Proc. IWAR’99, pp. 3-12.

[10] S. You, J. Hu, U. Neumann and P. Fox, “Urban Site Modeling From LiDAR”, Second International Workshop on Computer Graphics and Geometric Modeling CGGM 2003, Montreal, Canada, May 2003.

[11] E. Foxlin, and L. Naimak, “VIS-Tracker: A Wearable Vision-Inertial Self-Tracker”, Proc. of IEEE VR2003.

[12] Zhengyou Zhang, “Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting”, INRIA Research Report No. 2676, 1995.

[13] Y. Yokokohji, Y. Sugawara, and T. Yoshikawa, “Accurate Image Overlay on Video See-through HMDs Using Vision and Accelerometers”, Proc. IEEE Virtual Reality 2000, pp247-254.

[14] R. T. Azuma, “A Survey of Augmented Reality”, Presence: Teleoperators and Virtual Environments, vol. 6, 1997, pp355-385.

[15] T. Höllerer, S. Feiner, and J. Pavlik, “Situated Documentaries: Embedding Multimedia Presentations in the Real World”, Proc. of ISCW’99, pp79-86.

[16] W. Piekarski, B. Gunther, and B. Thomas, “Integrating Virtual and Augmented Realities in a Outdoor Application”, Proc. of IWAR’99, pp. 45-54

[17] H. Rehbiner, B. Ghosh, “Pose Estimation Using Line-Based Dynamic Vision and Inertial Sensors”, IEEE Transactions on Automatic Control, Vol 48, No. 2, February 2003.

[18] G. Klein, T. Drummond, “Robust Visual Tracking for Non-Instrumented Augmented Reality”, to appear in ISMAR 2003.

[19] V. Ferrairi, T. Tuytelaars, and L. Gool, “Markerless Augmented Reality with Real-time Affine Region Tracker”, ISAR 2002, pp. 87-96.

[20] K. Chia, A. Cheok, and S. Prince, “Online 6DOF Augmented Reality Registration from Natural Features”, In Proc. ISMAR 2002.

[21] D. Stricker, T. Kettenbach, “Real-time and Markerless Vision-based Tracking for Outdoor Augmented Reality Applications”, Proc. of ISMAR 2001.

[22] R. Behringer, J. Park, and V. Sundareswaran, “Model-based visual tracking for outdoor augmented reality”, Proc. ISMAR ’02

[23] OpenGL Programming Guilde, third version, ADDSION-WESLEY.