Abstract: A novel method for estimating vehicle roll, pitch and yaw using machine vision and inertial sensors is presented that is based on matching images captured from an on-vehicle camera to a rendered representation of the surrounding terrain obtained from an on-board map database. United States Geographical Survey Digital Elevation Maps (DEMs) were used to create a 3D topology map of the geography surrounding the vehicle, and it is assumed in this work that large segments of the surrounding terrain are visible, particularly the horizon lines. The horizon lines seen in the captured video from the vehicle are compared to the horizon lines obtained from a rendered geography, allowing absolute comparisons between rendered and actual scene in roll, pitch and yaw. A kinematic Kalman filter modeling an inertial navigation system then uses the scene matching to generate filtered estimates of orientation. Experiments using an instrumented vehicle operating at the test track of the Pennsylvania Transportation Institute were performed to check the validity of the method, and the results reveal a very close match between the vision-based estimates of orientation versus those from a high-quality GPS/INS system.

Keywords: Terrain Aided Localization, Inertial Navigation, Kalman Filter.

1. INTRODUCTION

The high sampling rates needed for vehicle chassis stability control, autonomous navigation and fault detection are primarily realized with the use of inertial sensors such as yaw rate gyros and lateral accelerometers. Because these sensors rely on integration to obtain position and orientation, biases can be introduced in these state estimates. These biases can grow unless external measurements of position and orientation, or assumptions about vehicle behavior, are applied.

Common methods to correct for the drift of inertial sensors include fusing this data with GPS measurements (Bevly [2004] and Ryu and Gerdes [2002]), and/or wheel speed sensor data (Dissanayake and Whyte [2001] and Chung and Borenstein [2001]. However, a novel corrector method is suggested by early missile and submarine guidance systems (Golden [1980], Denisim and Roberts [1989], Hostetler and Andreas [1983]) where external terrain features obtained by radar and sonar are compared to expectations or virtual observations from databases stored a priori. This technique is called terrain aided localization (Madhavan [2004]) and is the subject of this work.

Among sensors needed to observe terrain features for terrain aided localization of a vehicle, vision systems are a particularly compelling choice. First, visibility of key features within the surrounding scene can be assumed as a precondition for driving. Vision sensors readily identify features that impose stark geometric constraints on a world model, for example horizons, road edges and
markers, buildings, etc. and many of these features are intentionally placed for driving localization, e.g. lane markers. Vision sensors are relatively cheap compared to navigation-grade radar or GPS systems. The field of computer vision has made continuing improvements in faster image processing techniques and efficient image registration algorithms. These improvements, when combined with the readily available processing power of modern processors, can now allow comparisons to be completed in real-time. Finally, vision systems are already widely available in many production vehicles in applications ranging from lane detection to backup visibility assistance.

Additionally, information and the technology needed to measure, store, and retrieve terrain features has increasingly become available in the form of Digital Elevation Models (DEM), topographical maps and 3D models of the cities, for example see the works of Fruh and Zakhor [2001] and Grinstead and Abidi [2005] and USGS Website [2006]. This evolution in data storage and representation is motivated especially by portable and in-vehicle navigation aids, advanced 3D gaming, and high-speed retrieval and visualization of map databases, e.g. Google Earth, Microsoft’s Virtual Earth, TerraServer, NASA’s WorldWind, etc.

The notion of fusing sensor and map data for vehicle or robot localization is not new. For example, Edwards and Schoppmann [1988] describe the generation of a high resolution digital terrain map for autonomous ground vehicle navigation. With the public availability of United States Geographical Survey data, other approaches have used databases of stored DTMs and DEMs to aid in vehicle localization Talluri and Aggarwal [1992], Lerner and Rotstein [2004], Rodriguez [1990], Furst and Dickmanns [1999], Hoffman [1999]. For example, Talluri and Aggarwal [1992] take different views of the horizon and use them to search the underlying map for possible robot locations. In Lerner and Rotstein [2004], the pose of the camera at two consecutive frames is derived using a DTM and the corresponding features in two frames. In Rodriguez [1990], a map-based algorithm for passive aircraft navigation system is presented using stereo analysis on successive images to recover an elevation map which is matched to the reference digital map of the 3-D terrain, that in turn is used to determine the position and heading of the aircraft. In Furst and Dickmanns [1999], vision data is used along with a model of key landmarks (e.g. runway, buildings etc. around the airport) to aid in the positioning of the aircraft. In Hoffman [1999], semi-sparse terrain maps are generated and matched to get a vision-based state estimate, an estimate that is then fused with wheel odometry using a Kalman filter to further improve the state estimate. While the commonality of the above studies to the present work is evident, most of the above algorithms do not consider vehicle orientation or require a computational complexity that challenges real-time implementation.

Terrain-matching localization similar to this work has also been studied in the area of underwater vehicle localization, where x,y position and yaw orientation of the underwater vehicle are estimated by using a reference map and look-down sensors Zhang and Yang [2004]. This reference map is either a map generated using multi-beam echo sounder as in the work of Lucido and Zhang [1996], or from a previously measured Digital Elevation Map (DEM) as in the work of Strauss and Aldon [1999].

In many applications, knowledge of surrounding terrain is not known a-priori, yet localization of the vehicle may still be possible. For example, one can map apparent orientation and translation of the camera (and hence the vehicle carrying the camera) by comparing frame to frame any changes in the scene. There is a large body of research in this area including studies that fuse image and inertial sensor data (You and Neumann [2001], Rehbinder and Ghosh [2001], Strelow and Singh [2004], Hu and Uchimura [2004], Chroust and Vincze [2004]). However, nearly all map-free methods can suffer from an initial bias or long-term drift in the resulting state estimates due to the inherent lack of absolute references.

The work presented in this paper is an extension of terrain aided localization methods applied to vehicle localization during low-speed driving. As discussed above, previous algorithms for terrain aided localization are computationally expensive or have drift issues due to lack of an absolute reference. The method presented in this paper resolves those issues by using efficient curve matching techniques and an absolute terrain reference model. This method can be especially useful for estimating the attitude of the vehicle using a low cost camera and an IMU, as current methods of estimating vehicle yaw (e.g. GPS/IMU, multiple antenna GPS, gyro/compass) are either very expensive or imprecise. The monetary cost incurred using this method is comparable to a cheap GPS/IMU combination while providing much better estimates of orientation because of the availability of an absolute reference.

The remainder of this paper is organized as follows. Section 2 shows how to estimate camera movement if the correspondence between image features is known. Section 3 describes a method to generate the 3D representation of the environment. Section 4 develops an IMU measurement model including characterization of IMU error sources. Section 5 describes image alignment algorithms that remove error in the estimated
camera position. A kinematic Kalman filter is developed using vision and inertial measurements in Section 6. Experimental performance results are given showing comparison of state estimates to the results obtained by using GPS/INS Kalman filter in Section 7. A conclusion summarizes the main results of the paper.

2. OBTAINING CAMERA ROTATIONS FROM HORIZON DATA

To estimate the orientation of the vehicle in an outdoor unstructured environment by fusing measurements from vision and inertial sensors, the vehicle is assumed to be equipped with an Inertial Measurement Unit (IMU) and a single camera with its z-axis (a line passing through the image center and perpendicular to the image plane) aligned with the vehicle roll axis. A DEM of the surrounding terrain and an approximate estimate of the initial position and orientation of the vehicle is assumed to be provided. A template image is generated using a perspective projection model of the camera using the DEM data and estimates of position and orientation. Real images are compared to these template images using curve matching techniques explained in detail later to measure the difference in vehicle orientation in the real and rendering environment.

To find the best estimate (in the least square sense) of vehicle orientation, a relationship is needed between camera motion and corresponding feature changes in two images. In the mathematical transformation of the acquired data, rotating and translating a camera in a fixed world is equivalent to keeping the camera fixed and rotating and translating the world in the opposite direction. Under these assumptions, one can assume that the camera is stationary and apply a rotation and translation to the world points to emulate the camera motion. World and camera coordinate systems are assumed to be right hand rectangular coordinate systems. It is also assumed that the world points have already been converted into the camera coordinate system.

More formally, from kinematics of rigid bodies, if a point \( p_0 = (x_0, y_0, z_0) \) at time \( t_0 \) moves to a location \( p_1 = (x_1, y_1, z_1) \) at time \( t_1 \) as a result of the motion of a rigid body, the following relation between \( p_0 \) and \( p_1 \) applies after making a small angle approximation (\( \cos(\angle) \approx 1, \sin(\angle) \approx \angle \)):

\[
\begin{pmatrix}
  x_1 \\
  y_1 \\
  z_1
\end{pmatrix} = \begin{bmatrix}
  1 & \delta_z & -\delta_y \\
  -\delta_z & 1 & \delta_x \\
  \delta_y & -\delta_x & 1
\end{bmatrix} \begin{pmatrix}
  x_0 \\
  y_0 \\
  z_0
\end{pmatrix} + \begin{pmatrix}
  \delta_x \\
  \delta_y \\
  \delta_z
\end{pmatrix}
\]  

The perspective projection model of a camera is given by

\[ u = \lambda \frac{x_0}{z_0}, \quad v = \lambda \frac{y_0}{z_0} \]  

Here \((u, v)\) are the image coordinates of point \( p_0 \). Using this model, the image coordinates \((U, V)\) of the point \( p_1(x_1, y_1, z_1) \) are given by

\[ U = \lambda \frac{x_1}{z_1}, \quad V = \lambda \frac{y_1}{z_1} \]

Using values of \( x_1 \), \( y_1 \) and \( z_1 \) from equation (1), we get

\[ U = \lambda \frac{x_0 + \delta_x y_0 - \delta_y x_0 + \delta_z}{z_0 + \delta_z}, \quad V = \lambda \frac{-\delta_y x_0 + y_0 + \delta_x z_0 + \delta_z}{z_0 + \delta_z} \]

Because the features being observed are very distant compared to the translational distance of the camera, we can safely assume that \( \frac{\delta_x}{\delta_z} \approx 0 \), \( \frac{\delta_y}{\delta_z} \approx 0 \), and \( \frac{\delta_z}{\delta_z} \approx 0 \). Rearranging and grouping terms associated with the camera orientation parameters \((\delta_x, \delta_y, \delta_z)\), we obtain (5).

\[
\begin{bmatrix}
  \lambda U & \lambda V & -U \ast u - V \ast v
\end{bmatrix} \begin{bmatrix}
  \delta_x \\
  \delta_y \\
  \delta_z
\end{bmatrix} = V \ast u - U \ast v \]  

We can combine \( N \) such equations into a matrix equation (6).

\[ A \delta = B \]

\[
A = \begin{bmatrix}
  \lambda U_1 & \lambda V_1 & -U_1 \ast u_1 - V_1 \ast v_1 \\
  \lambda U_2 & \lambda V_2 & -U_2 \ast u_2 - V_2 \ast v_2 \\
  \vdots & \vdots & \vdots \\
  \lambda U_N & \lambda V_N & -U_N \ast u_N - V_N \ast v_N
\end{bmatrix}, \quad B = \begin{bmatrix}
  V_1 \ast u_1 - U_1 \ast v_1 \\
  V_2 \ast u_2 - U_2 \ast v_2 \\
  \vdots \\
  V_N \ast u_N - U_N \ast v_N
\end{bmatrix}
\]

This matrix equation is a system of linear equations with more equations than unknowns, i.e. an over-determined system which can be solved using the pseudo inverse method to get a least squares error estimate. If we assume that \((A') A\) is non-singular, then the equation (6) can be solved using least squares as

\[ \delta = (A' A)^{-1} A' B \]

The term \((A') A\) will be non-singular if \( u_1 \) and \( v_1 \) do not lie on a straight line passing through the origin.

3. TERRAIN REPRESENTATION AND RENDERING

The vision-inertial Kalman filter requires a 3-D representation of the terrain around the test-site. All data used in the experiments of this study were collected at the Pennsylvania Transportation...
In terms of true vehicle rotation rate $r$, the following equation gives gyro output $r_{gyro}$ by a random constant. Hence, the exponentially correlated noise is very high (approx. 11 hrs [2004]). For RLGs, the time constant for exponential drift component (Bevly [2004], Egziabher [2004], Brown and Hwang [1992], Rogers [2000]). The bias drift was modeled as a random constant. The variance of the random constant can be derived using the bias-instability ($\sigma_B$). The differential equation governing the process is given as:

$$\dot{b}_{gyro} = 0$$

$$\sigma^2_{b_{gyro}} = \sigma^2_B$$

The wide-band noise is assumed to be normally distributed with zero mean and a variance as given by equation 10. The variance of this wide-band noise is denoted by ($\sigma_{gyro}$).

$$E[w_{gyro}^2] = \sigma^2_{gyro}$$

5. ORIENTATION ESTIMATES USING VISION: HORIZON MATCHING

To complement measurements from inertial sensors, orientation measurements are also obtained by comparing real and rendered images. As shown in section 2, we can estimate the deviations in roll, pitch and yaw of the vehicle by comparing real and rendered images if we know the feature correspondences in the two images. To obtain feature correspondences, the horizon curves extracted from the real and rendered images are matched. Curve matching techniques have been used extensively for image/map correspondence Talluri and Aggarwal [1992], Ernst and Flinchbaugh [1989], Freeman and Morse [1967], Rodriguez [1990]. Refinements have also been studied that address map-specific features; for example a curve matching algorithm is described in Wollson [1990], Schwartz [1987] which is based on matching high curvature points along the curve length. Representing the curves by characteristic strings of high curvature points essentially assigns them shape signatures which are translation and rotation invariant. These shape signatures are then compared to match the curves or to find the longest matching sub-curve between the two curves. This method was tested in this study and found to work well for matching of two rendered horizon curves, but lacked robustness when matching a rendered horizon to a real horizon curve. Investigation is ongoing to make this algorithm more robust for matching rendered and real horizon curves.

To match rendered horizon curves with real horizon curves, a Random Sample Grid Search (RSAGS) technique was used. This method simplifies the iterative curve-matching procedure by...
avoiding the use of the entire rendered horizon-line data set. Instead, only a very small subset of the of the horizon-line data is used to generate an estimate of camera orientation deviations for curve matching. In this study, four points are selected uniformly and randomly from the rendered curve. A transformation given by equation 11 is applied to these points for a grid of parameters \((t_u, t_v, \alpha)\) around the current curve position. In equation 11, \(t_u\) denotes translation in \(u\)-direction, \(t_v\) denotes translation in \(v\)-direction and \(\alpha\) denotes rotation of the curve in the plane of the image. \(W\) and \(H\) denote the width and height of the image in pixels respectively. \((u, v)\) denote the image coordinates of the random points and \((u_t, v_t)\) denote the image coordinates of the random points after the transformation. This transformation signifies a Euclidian transformation, which rotates the image pixels by \(\alpha\) about the image center and then translates them by \(t_u\) and \(t_v\) in image \(u\) and \(v\) coordinate directions respectively.

\[
\begin{bmatrix}
  u_t \\
  v_t \\
  1
\end{bmatrix} =
\begin{bmatrix}
  1 & 0 & t_u + W/2 & 0 & 1 & t_v + H/2 & 0 & 0 & 1 \\
  0 & 1 & +W/2 & 0 & 1 & +H/2 & 0 & 0 & 1
\end{bmatrix} 
\begin{bmatrix}
  \cos(\alpha) & -\sin(\alpha) & 0 \\
  \sin(\alpha) & \cos(\alpha) & 0 \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  u \\
  v \\
  1
\end{bmatrix}
\] (11)

For each point on the parameter grid \((t_u, t_v, \alpha)\), the sum of squared vertical distance is calculated between the transformed points and the points on the real curve having the same \(u\)-coordinate. In Figure 1, \(d_1, d_2, d_3\) and \(d_4\) denote these vertical distances. The set of parameters which minimizes the sum of squared vertical distances \(d_1^2 + d_2^2 + d_3^2 + d_4^2\) is taken as the best-fit parameter set. Using this set of best-fit parameters, the transformation given by equation 11 is applied to all the points in the rendered curve, and from all points, a sum of squared vertical distances from the real curve is calculated (denoted as \(SSV D_1\)). To be robust against the noise in image data, this entire process is repeated on several sets of four random points. The transformation giving the minimum sum of squared vertical distance \(SSV D_{min}\) is thereafter assumed to be the best estimate of match parameter set \((t_{uest}, t_{vest}, \alphaest)\). This estimate provides the pixel correspondences in the two curves, and from this correspondence, equation 7 is used to find the estimates of roll, pitch and yaw.

The size of the parameter grid to be searched for RSAGS is determined by the rate at which images are taken. The higher the frame rate, the smaller the grid size that has to be searched due to the smaller total motion of the image from frame to frame. The resolution of the grid determines the accuracy of pixel correspondence. In this study, a frame rate of 29.97 frames per second and a grid resolution of 1 pixel for \(t_u\) and \(t_v\) and 0.1 degrees for \(\alpha\) was used. The search grid size was taken to be 20 pixels for \(t_u\) and \(t_v\) and 2 degrees for \(\alpha\).

To confirm the validity of the RSAGS curve-fitting method, a numerical study was first conducted by comparing two rendered horizon curves. To create the two curves, we started with one horizon and moved the camera by small angles to generate another horizon by a camera perspective projection model. Hence, the true deviation of the camera orientation was known. The RSAGS algorithm was used to find the best-fit translation and rotation that matches these two horizon curves, and Figure 2 shows the matching results. Here \((\phi_1, \theta_1, \psi_1)\) denote the true camera motion parameters in roll, pitch and yaw directions, and \((\phi_m, \theta_m, \psi_m)\) denote the estimated camera motion parameters. There is a good agreement between true and estimated camera motion parameters. For comparison, the figure also shows a horizon curve generated using estimated camera motion parameters.

As the next validation step, a real image of a horizon was captured from a stationary vehicle at the PSU Pennsylvania Transportation Institute Test Track using standard contrast threshold techniques to find the horizon boundary between sky/land. This horizon was compared to a rendered horizon curve obtained by using the local DEM and the vehicle’s measured DGPS position and IMU orientation. Figure 2 shows the RSAGS results. Because no truth value of camera orientation deviation in real and rendered environment is available, deviations in camera orientation \((\phi_1, \theta_1, \psi_1)\) are estimated by using RSAGS. These deviations give an estimate of error in camera orientation predicted by using an IMU alone. To show a comparison of the match between rendered and measured horizon, the horizon curve generated by the renderer after moving the camera through the estimated deviations is also shown in Figure 2.

The horizon-only match does little justice to the quality of agreement between the real and rendered scene. To illustrate more fully, Figure 3 shows an overlay of real and rendered 3-D images showing matching of the horizon feature from test-track images.
6. FUSING VISION MEASUREMENTS WITH INERTIAL DATA

After obtaining vehicle orientation measurements from inertial and vision sensors, an estimation algorithm is needed which uses these measurements along with their error characteristics to generate a best possible orientation estimate. A kinematic Kalman filter is used to fuse vision and inertial data by using vision measurements to correct inertial integration errors as well as to estimate biases in inertial sensors. An advantage of a kinematic estimator, as opposed to dynamic estimator, is that there is no estimation error associated with system modeling error.

Euler angles and vehicle body rates \((\hat{u}_\phi, \hat{u}_\theta, \hat{u}_\psi)\) assuming roll \((\phi)\) and pitch \((\theta)\) angles to be small \((< 5 \text{ deg})\) is given by:

\[
\begin{bmatrix}
\dot{\phi} \\
\dot{\theta} \\
\dot{\psi}
\end{bmatrix} =
\begin{bmatrix}
1 & 0 & \phi \\
0 & 1 + \phi & -\phi \\
0 & \phi & 1
\end{bmatrix}
\begin{bmatrix}
\hat{u}_\phi \\
\hat{u}_\theta \\
\hat{u}_\psi
\end{bmatrix}
\]  \hspace{1cm} (12)

Using equation 9, we get the following state space form for gyro biases.

\[
\begin{bmatrix}
\hat{b}_{\phi} \\
\hat{b}_{\theta} \\
\hat{b}_{\psi}
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\omega_x \\
\omega_y \\
\omega_z
\end{bmatrix}
\]  \hspace{1cm} (13)

Using equations 12 and 13 and taking into account the effect of gyro biases and process noise, the state space form of the estimator can be written as:

\[
\begin{bmatrix}
\dot{\phi} \\
\dot{\theta} \\
\dot{\psi}
\end{bmatrix} =
\begin{bmatrix}
\phi \\
\theta \\
\psi
\end{bmatrix} +
\begin{bmatrix}
1 & 0 & \phi \\
0 & 1 + \phi & -\phi \\
0 & \phi & 1
\end{bmatrix}
\begin{bmatrix}
\hat{u}_\phi - b_{\phi} \\
\hat{u}_\theta - b_{\theta} \\
\hat{u}_\psi - b_{\psi}
\end{bmatrix}
\]  \hspace{1cm} (14)

Here \(w_{\text{gyro}}\) is the process noise with covariance given as

\[
E[w_{\text{gyro}}^2] = Q_c
\]  \hspace{1cm} (15)

Roll, pitch and yaw angle measurements are directly available using comparison of real and rendered images. So, the output equation can be written as

\[
\begin{bmatrix}
\phi_e \\
\theta_e \\
\psi_e
\end{bmatrix} =
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\phi \\
\theta \\
\psi
\end{bmatrix} +
\begin{bmatrix}
\omega_x \\
\omega_y \\
\omega_z
\end{bmatrix}
\]  \hspace{1cm} (16)

Here \(v\) is the measurement noise with covariance:

\[
E[v^2] = R_v
\]  \hspace{1cm} (17)

where the covariance of the measurement depends on the error covariance of the curve matching algorithm.

The continuous time state space model given by equations 14 and 16 is converted to a discrete-time state space model of the form given by equation 18 using a forward transformation rule \((\Phi = I + AT_s, \Gamma = B, T_s, H = C_i)\), where \(T_s\) is the sampling rate. The measurement model is given by equation 19.

\[
\hat{x}_k = \Phi \hat{x}_{k-1} + \Gamma u_{k-1} + w_{k-1}, \quad w_k \sim N(0, Q_k)
\]  \hspace{1cm} (18)

\[
z_k = H_k x_k + v_k, \quad v_k \sim N(0, R_k)
\]  \hspace{1cm} (19)
where $\Phi$ is the discretized state matrix ($A$), $\Gamma$ is discretized input matrix ($B_u$) and $H_k$ is discretized output matrix ($C$).

A linear discrete Kalman filter is used which follows a predictor-corrector form and is comprised of a time update and a measurement update. The predictor-corrector form of a Kalman filter is very well described in Gelb [1974]. The time update is given by

$$\hat{x}_k(+) = \Phi \hat{x}_{k-1}(+) + \Gamma u_{k-1}$$
$$P_k(-) = \Phi_{k-1}P_{k-1}(+) \Phi^T_{k-1} + Q_{k-1}$$

An important requirement for the Kalman filter to work is that the image and inertial data should be aligned in time. Because the inertial data is sampled much more quickly than image data, there are measurement cases where the vision measurement is not available. At these times, equation 20 is repeatedly used to predict the state of the system by using $x_k(+) = x_k(-)$ and $P_k(+) = P_k(-)$ for the next iteration. When a vision measurement is available, a measurement update step is applied to the states as given in equation 21

$$K_k = P_k(-)H^T_k [H_k P_k(-) H^T_k + R_k]^{-1}$$
$$\hat{x}_k(+) = \hat{x}_k(-) + K_k [z_k - H_k \hat{x}_k(-)]$$
$$P_k(+) = [I - K_k H_k] P_k(-)$$

In the next section, this Kalman filter is implemented using experimental data collected at the Pennsylvania Transportation Institute’s test track facility on a Mercury Tracer station wagon vehicle instrumented with the IMU described previously and a windshield-mounted camcorder.

### 7. EXPERIMENTAL RESULTS

Experiments were conducted where gyro data from the IMU as well as the video of horizon features recorded from an on-vehicle camera were used off-line in a Kalman filter to predict roll, pitch and yaw angles. Orientation information from a military grade GPS/INS system was also recorded as a reference.

As shown in Figures 5, 6, and 7, the roll, pitch, and yaw angles estimated by vision/INS Kalman filter agree very well with the estimate from the GPS/INS system. The distribution of deviations between these two estimators in the roll and pitch estimates (not shown) follows a Gaussian distribution with a mean of 0.0034 degrees and a 2$\sigma$ value of 0.33 degrees for roll, a mean of 0.0382 degrees and a 2$\sigma$ value of 0.22 degrees for pitch, but a roughly uniform variation between -2 to 2.5 degrees in yaw (the yaw issue is discussed below). The figures also show that after around 20 to 30 seconds, the roll, pitch, and yaw gyro bias stabilizes at around -0.04 deg/s, 0.035 deg/s, and 0.06 deg/s respectively. This shows that the gyro biases are indeed constant as assumed in the gyro model.
can increase quite fast if the change of acceleration of the vehicle is small. As the vehicle in this study had very small accelerations - it was driven at a nearly constant speed of 5 mph and over sweeping turns during our experiment - there might be an error in the yaw estimate of the GPS/INS system. The vision/INS Kalman filter detects a yaw gyro bias as shown in Figure 7, which stabilizes to a nearly constant value of 0.06 deg/s after around 30 sec. If indeed there is an error in detecting the yaw gyro bias in the GPS/INS system, it will give a difference in yaw estimate similar to what was observed in the data. Further investigation to isolate the cause of this discrepancy is ongoing.

8. CONCLUSIONS

This work demonstrated the use of horizon lines detected by a camera and DEM data stored a priori to measure vehicle roll, pitch and yaw angles. The horizon lines seen in the captured video were compared to the horizon lines generated from the rendered geography using curve matching techniques to generate the deviations of vehicle roll, pitch and yaw angles in real and rendered environments. A kinematic Kalman filter implemented using inertial and vision data was used to provide estimates of roll, pitch and yaw of the vehicle. Comparison of these estimates with measurements from a high quality GPS/INS system showed good agreement in roll and pitch angles. If the roll and pitch angles given by GPS/INS system are assumed to be the true roll and pitch angles, the vision/IMU Kalman filter implemented in this paper was able to estimate roll and pitch angles with an error bound (2σ) of 0.33 degrees and 0.22 degrees respectively. The yaw angle estimates differ by what seems to be a different yaw gyro bias estimate between the estimator described in this work versus the one operating within the GPS/IMU system. In other words, we suspect that the vision/IMU system developed in this work is providing better absolute yaw estimates than the best GPS/IMU system we were able to obtain commercially.