Localization Through Fusion of Discrete and Continuous Epipolar Geometry with Wheel and IMU Odometry

Jinglin Shen, David Tick and Nicholas Gans

Abstract—This paper describes a novel sensor fusion implementation to improve the accuracy of robot localization by combining multiple visual odometry approaches with wheel and IMU odometry. Discrete and continuous Homography Matrices are used to recover position, orientation, and velocity from image sequences of tracked feature points. An Inertial Measurement Unit (IMU) and wheel encoders also measure linear and angular velocity of mobile robot. A Kalman filter fuses the measurements from the visual and inertial measurement systems. Time varying matrices in the Kalman filter allow each sensor to receive higher or lower weight in situations where each is more or less accurate. Experiments are performed with a camera and an IMU (Wiimote controller) mounted on a mobile robot.

I. INTRODUCTION

Building mobile robotic systems that are capable of real-time autonomous navigation is a complex, multi-faceted problem. One of the primary aspects of this problem is the task of localization. Localization, or pose-estimation, refers to the task of estimating the position, orientation and velocity (both angular and linear) of the robot over time. There are many established ways to approach the task of localization including wheel odometry [1], inertial sensors [1], GPS [1], sonar [2], and IR/laser-based range finding sensors [3].

There have been significant results in localization techniques based solely on vision data. Visual odometry is a method of localization that uses one or more cameras to constantly capture images or video frames, taken of a scene [4]. The frames are analyzed sequentially using various computer vision techniques. The analysis of these frames estimates angular and linear velocities of the camera between each time step. These velocities are integrated over time to estimate how the camera has moved.

Some vision-based localization techniques are designed to calculate the pose of the camera relative to a well known reference object that appears in each frame [5], [6]. These techniques are usually dependent on accurate prior knowledge regarding the geometric properties of the scene or a reference object. In many situations, there exists little accurate prior knowledge regarding a scene and the objects therein. However, in such cases, two frames can be compared with one another based on a set of feature points which exists in both frames. Once a set of feature points is identified in both frames, a mapping of the feature points from one frame onto the next must exist. This mapping can be described as a linear transformation and encapsulates the rotation and translation of the camera that occurs between the taking of each picture. In practice, the Essential matrix or Euclidean Homography Matrix is used to estimate a camera’s pose in terms of a set of rotational and translational transformations [7], [8].

Over the years, many control systems have been designed that utilize Essential or Homography Matrices in vision-based robotic tasks [9]–[12]. The Homography Matrix, in its discrete form, has been widely used for many years by the community [13]–[17]. Of particular interest is the application of the Euclidean Homography Matrix as a means of estimating velocity [18]. However, it seems that there has not been as much work done regarding applications of the Homography Matrix in its continuous form.

The Euclidean Homography Matrix can be used in both a continuous as well as a discrete form [8]. If one does a discrete homography-based estimation of the camera position and orientation (i.e. pose), then one can also integrate the continuous estimate of the velocity at each time step to extract the pose as well. Discounting the effects of noise, the pose estimate obtained from integrating the continuous homographic estimate of angular velocity must agree with the discretely estimated position of the camera. This notion was initially explored in [19], and verified with simulations.

A pose estimate produced in this manner can be further enhanced by combining visual odometry with traditional wheel odometry and other sensor devices. This can be achieved through various signal processing techniques known as sensor fusion. The Kalman filter is particularly useful for fusing signals from multiple sensors and removing errors in localization that occur due to many factors such as sensor noise, quantization, flawed process modeling, and sensor bias or drift [1], [20].

Vision-based Kalman filter approaches have been studied over the years and used in robot localization and mapping problems. Soatto et al. give two Kalman filter approach for motion estimation based on the Essential matrix [21]. Some other methods use image features as inputs to the Kalman filter with pose and/or velocity as output, which are more computational efficient and avoid the restrictions and limitations of the pose reconstruction methods [22], [23]. Zhang et al. use a trinocular stereo system mounted on a mobile robot to build a local map about the environment while the robot is navigating [24]. A Kalman filter is used to merge matched line segments from the trinocular stereo system. In [25], a multisensor, on-the-fly, localization system
is proposed to fuse estimation from visual sensors with other types of sensors, such as laser range finders. Furthermore, Gutmann developed a vision-based Markov-Kalman method for robots observing known landmarks, which combines the accuracy of Kalman filter and robustness of Markov method [26].

In this paper, we propose a novel localization technique involving vision-based estimates of pose and velocity from continuous and discrete Homography matrices, an IMU, and wheel encoders. The visual odometry method uses a single camera rigidly mounted on the robot. The change in camera pose is estimated by using the discrete form of the Euclidean Homography Matrix, while the continuous form of the Euclidean Homography Matrix is utilized to estimate the velocity of the camera over time. The IMU measures angular velocity, and the wheel encoders of the mobile robot measure linear and angular velocity. The system utilizes a Kalman filter to fuse all the estimates and remove error which accrues over time due to integration, noise, and quantization [20]. Furthermore, a low-cost IMU is used, specifically a Nintendo Wiimote. The results obtained, despite the poor performance of the IMU, show the strength of the system and the possibility of low-cost alternatives for localization. Some sensors are more accurate in certain regimes of operation. The proposed system alters the covariance matrices used in the Kalman filter based on the system state. This allows sensors that are currently more accurate to receive a higher weighting in the sensor fusion. This results in a more accurate estimate and allows for improved estimation of bias on certain sensors.

Section II will clarify the terminology used in this paper as well as formally define and model the problem. Section III will then explain the proposed sensor fusion approach. We present experiment results in section IV that illustrate the effectiveness of the proposed system. Finally, in section V we will espouse the conclusions that reached from this work.

II. BACKGROUND

Different conventions exist for the notation and terminology used in the vision-based estimation and robot control. For example, the vision community typically assigns a Cartesian reference frame to a camera with the z-axis oriented along the optical axis, while the robotics community typically assigns a reference frame to a wheeled mobile robot with the z-axis oriented perpendicular to the plane of motion. In this paper, we follow the separate conventions for the vision system and mobile robot when discussing the development of each. The coordinate system of the vision-system is mapped to the frame of the robot through a constant rotation.

A. Formal Definition and Terminology

Navigation sensors, including cameras, make measurements with respect to a moving body frame, here called $\mathcal{F}_b$ for camera body frame and $\mathcal{F}_r$ for robot body frame. These measurements are then rotated to obtain localization with respect to the world frame, here called $\mathcal{F}_w$. In this paper, the orientation and position (collectively referred to as pose) of a body frame in terms of the world frame is expressed as a function of time by including the index number for a given time step in parentheses. For example, in Fig. 1 $\mathcal{F}_b(t_0)$ describes the pose of the body frame $\mathcal{F}_b$ attached to a camera at time index $t_0$ as measured from $\mathcal{F}_b(t_0)$, while $\mathcal{F}_b(t_k)$ describes the pose of $\mathcal{F}_b$ at time $t_k$ as measured from $\mathcal{F}_b(t_0)$.

The changes in the pose of $\mathcal{F}_b$ that occur over the time interval $[t_0, t_1, \ldots, t_{k-1}, t_k]$ is described in terms of a translation vector $T_k \in \mathbb{R}^3$ and a rotation matrix $R_k \in SO(3)$. At any given time $t_k$, the instantaneous linear and angular velocities of the body frame $\mathcal{F}_b$ are described as a pair of vectors $(v_k, \omega_k)$, where $v_k \in \mathbb{R}^3$ and $\omega_k \in \mathbb{R}^3$. We will use subscript $k$ to denote time step $t_k$ also for other variables in the rest of the paper.

In this work, a camera is rigidly attached to a wheeled mobile robot. The Cartesian reference frame of the camera, $\mathcal{F}_b$, is oriented such that the z-axis is oriented along the optical axis, the x-axis is oriented along the horizontal direction of the image plane, and the y-axis is oriented parallel to the vertical direction of the image. This is illustrated in Fig. 1. The reference frame of the robot, $\mathcal{F}_r$, is attached to the robot’s center of rotation, with x-axis aligned along the robot’s heading, y-axis oriented to the left of the robot along the horizontal direction, and z-axis oriented upwards along vertical direction. Similar to the camera frame, $\mathcal{F}_r(t_k)$ denotes the pose of $\mathcal{F}_r$ at time $t_k$. This is illustrated in Fig. 2.

Without loss of generality, the origin of $\mathcal{F}_b$ and $\mathcal{F}_r$ are the same. As discussed, the z-axis of $\mathcal{F}_b$ is coincident with the x-axis of $\mathcal{F}_r$, and the x-axis of $\mathcal{F}_b$ is coincident with the y-axis of $\mathcal{F}_r$. Thus they are separated by a constant rotation $R_{br}$. An IMU is rigidly attached to the robot, with its axes aligned with $\mathcal{F}_r$. Only rotations are measured by the IMU in this work. In this paper, the static world frame $\mathcal{F}_w$ is defined as the initial pose of the robot, i.e., $\mathcal{F}_w = \mathcal{F}_r(t_0)$.

B. Robot, Wheel Encoders and IMU

The wheeled robot follows the kinematic unicycle model [27] and moves in a plane spanning the x and y axes of $\mathcal{F}_w$ and $\mathcal{F}_r$. It has two degrees of freedom. It can rotate about the z-axis of $\mathcal{F}_r$ with an angular velocity $\omega_{zk}$ and translate along the x-axis with a linear velocity $v_{zk}$ with respect to $\mathcal{F}_r$. The location of the robot in $\mathcal{F}_w$ is the origin of $\mathcal{F}_w(t_k)$, given by $[x_{pk}, y_{pk}, 0]^T$. The orientation of the robot is $\theta_k$, and is measured by the angle between the x-axes of $\mathcal{F}_w$ and $\mathcal{F}_r$. This is illustrated in Fig. 2.

Wheel encoders are used to measure the angular velocity of each wheel and can provide information for odometry and localization. The angular velocity of the left and right wheel are given by $\omega_{lk}$ and $\omega_{rk}$ respectively, as shown in Fig. 2. These measurements are converted into linear velocity of the
wheels by
\[ v_L = \omega_L \times r \]
\[ v_R = \omega_R \times r \]
where \( r \in \mathbb{R}^+ \) is the constant radius of the wheels. Denoting the constant distance between the wheels as \( b \in \mathbb{R}^+ \), the linear and rotational velocity of the robot is given by

\[ v_x = \frac{v_L + v_R}{2} \quad \omega_x = \frac{v_L - v_R}{b} \]

By numerically integrating the estimates of \( v_{\omega x} \) and \( \omega_{\omega x} \) over time, the pose estimate of the robot in \( F_w \) is given by

\[
\begin{bmatrix}
  x_p_k \\
  y_p_k \\
  \theta_k
\end{bmatrix} = \sum_{i=t_0}^{t_k} \begin{bmatrix}
  -\sin \theta_i \Delta t & 0 & v_{\omega x} \\
  \cos \theta_i \Delta t & 0 & 0 \\
  0 & 1 & \Delta t
\end{bmatrix} \begin{bmatrix}
  v_{\omega x} \\
  \omega_{\omega x}
\end{bmatrix}
\]

where \( \Delta t \in \mathbb{R}^+ \) is the time difference between two time steps.

The IMU detects linear acceleration using accelerometers, and detects angular velocities using gyroscopes. Due to the limited motion of the robot, we use only one of the gyroscopes to measure angular velocity \( \omega_z \). The orientation of the robot, as estimated using the rate gyros, is given by

\[ \theta_k = \sum_{i=t_0}^{t_k} \omega_{\omega x} \Delta t. \]

C. Pinhole Camera Model

Fig. 3 illustrates a camera taking two images from two different poses \( F^*_b \) and \( F_b(t_k) \). \( F^*_b \) is considered a static reference frame, such as the pose at time \( t = 0 \). \( F_b(t_k) \) is considered a moving frame or current frame. The changes which exist between the two poses are encapsulated by \( (T_k, R_k) \). Fig. 3 also shows a collection of feature points that all lie in the plane \( \pi \). The 3D coordinates of each feature point, as measured from the reference frame \( F_b(t_k) \), are defined in terms of constant vectors \( \overline{m}_j \in \mathbb{R}^3 \). Similarly, the 3D coordinates of each feature point as measured from \( F^*_b \) are defined in terms of constant vectors \( \overline{m}_j^* \in \mathbb{R}^3 \). Formally these vectors are given as

\[ \overline{m}_j^* \in \mathbb{R}^3 = [x_j^*, y_j^*, z_j^*]^T, \forall j \in \{1, \ldots, N\} \]
\[ \overline{m}_j(t) \in \mathbb{R}^3 = [x_j(t), y_j(t), z_j(t)]^T, \forall j \in \{1, \ldots, N\}. \]

Two images are captured by the camera at the two poses \( F^*_b \) and \( F_b(t_k) \). This is modeled by projecting the feature points onto the 2D image planes. The coordinates of the feature points in these 2D planes are expressed as a normalized set of 3D coordinates, where depth along the \( z \)-axis is normalized to one. The normalized image plane coordinates,
as measured from $\mathcal{F}_b(t_k)$ are defined as vectors $m_j(t_k) \in \mathbb{R}^3$. Similarly, the normalized image plane coordinates of each feature point measured from $\mathcal{F}_b$ are vectors $\bar{m}_j \in \mathbb{R}^3$. These vectors are expressed as

$$m^*_j \in \mathbb{R}^3 = \begin{bmatrix} x^*_j \\ y^*_j \\ z^*_j \\ 1 \end{bmatrix}, \forall j \in \{1, \ldots, N\}$$

$$m_j(t) \in \mathbb{R}^3 = \begin{bmatrix} x_j(t) \\ y_j(t) \\ z_j(t) \\ 1 \end{bmatrix}, \forall j \in \{1, \ldots, N\}.$$

D. Discrete and Continuous Euclidean Homography

In this work, a visual odometry method is developed from the discrete and continuous Euclidean Homography matrices. This gives a vision-based estimate of position, orientation, and linear and angular velocities. While the discrete homography has been used extensively, the continuous homography seems to attract less attention.

Both planar homography cases assume all feature points are coplanar\(^1\). Consider two images of points $\mathbf{m}(t_k)$ on plane $\pi_c$. The transformation between the two images is given by [7], [8]

$$\mathbf{m}_j = \mathbf{R} \mathbf{m}_j^* + T. \quad (3)$$

The relationship in (3) can be rewritten in terms of image points as

$$m_j = (R + \frac{1}{d^*}TN^*)m^*_j$$

where $n^* = [n^*_x, n^*_y, n^*_z]^T$ is the unit constant normal vector of plane $\pi_c$, measured in $\mathcal{F}_b^*$, and $d^*$ is the constant scalar distance from the optical center of the camera (i.e. the origin of $\mathcal{F}_b^*$) to the plane $\pi_c$. We assume that $d^*$ is known in this initial investigation\(^2\).

The matrix

$$H_d = R + \frac{1}{d^*}TN^*$$

is the well known Euclidean Homography Matrix, which we will refer to as the discrete Homography Matrix. Given four or more points, the matrix $H_d(t_k)$ can be recovered and used to estimate the translation vector $T_k$ and the rotation matrix $R_k$ [7].

In the continuous case, image point $m_j(t_k)$ and its optical flow $\dot{m}_j(t_k)$ are measured rather than the image pair $m^*_j$ and $m_j(t_k)$. The time derivative of $\mathbf{m}_j(t_k)$ satisfies

$$\dot{m}_j = \hat{\omega}m_j + v \quad (4)$$

where $\hat{\omega}(t_k)$ is the skew-symmetric matrix of $\omega(t_k)$.

The relationship in (4) can be rewritten in terms of image points as [8]

$$\dot{m}_j = (\hat{\omega} + \frac{1}{d^*}vn^*T)m_j - \frac{\dot{z}_j}{z_j}.$$
process noise covariance matrix $Q_k \in \mathbb{R}^{6 \times 6}$ is defined as

$$Q_k = \begin{bmatrix}
\sigma_3^\omega & 0 & \sigma_2^\omega & 0 & 0 & 0 \\
0 & \sigma_1^\beta & 0 & 0 & 0 & 0 \\
\sigma_2^\omega & 0 & \sigma_1^\omega & 0 & 0 & 0 \\
0 & 0 & 0 & \sigma_3^\delta & \cos \theta_k \sigma_2^\omega & \sin \theta_k \sigma_3^\omega \\
0 & 0 & 0 & \sigma_1^\omega & \sigma_3^\delta & 0 \\
0 & 0 & 0 & \cos \theta_k \sigma_1^\omega & \sin \theta_k \sigma_3^\omega & \sigma_2^\omega \\
\end{bmatrix}$$

The process noise is designed as random walks between the angular velocity and orientation, and linear velocity and position. According to the random walk model, the change of $\omega$ is a tunable constant reflecting the expected amount of change in $\omega$. The terms $\sigma_i^\omega$ are defined in the same manner for linear velocity and bias.

In system equation (6), the measurement vector $y_k$ is defined as

$$y_k = [\omega_w, \omega_e, v_w, v_c, \omega_k, \theta_k, x_d, y_d]^T \in \mathbb{R}^8. \quad (7)$$

In (7), $\omega_w$, $\omega_e$, $\omega_k$ are measurement of angular velocity $\omega_k$ from the wheel encoders in (2), rate gyro, and the continuous Homography respectively. The terms $v_w$ and $v_c$ are measurement of linear velocity $v_k$ from wheel encoders and continuous Homography. Since different conventions were used to define camera and robot frame, all measurements made from the vision-based estimation in $y_k$ are converted from the camera frame to the robot frame by the constant rotation matrix $R_{br}$. The measurement matrix $H_k \in (R)^{8 \times 6}$ is defined accordingly as

$$H_k = \begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
\end{bmatrix}.$$  

The measurement noise covariance matrix $R_k \in \mathbb{R}^{8 \times 8}$ is a diagonal matrix. For simplicity, the diagonal elements are listed as a vector

$$\text{diagonal}(R_k) = [\sigma^\omega, \sigma^\rho, \sigma^w, \sigma^c, \sigma^s, \sigma^d, \sigma^y] \in \mathbb{R}^8.$$  

Rather than keep all terms in $R_k$ constant, as in many works using the Kalman filter, two terms are changed according to the motion of the robot. Namely, $\sigma^\omega$ and $\sigma^\rho$ are defined as

$$\sigma^\omega = (\delta^\omega_w \ast \Delta_t \ast \omega_w) + (\delta^\omega_e \ast \Delta_t \ast v_w) \quad (8)$$  

$$\sigma^\rho = (\delta^\rho_w \ast \Delta_t \ast \omega_w) + (\delta^\rho_c \ast \Delta_t \ast v_w) \quad (9)$$

where $\delta^\omega_w$ and $\delta^\rho_w$ are some constants. The other $\sigma$ terms are constants tuned according to the variance of each measurement.

The value of diagonal elements in $R_k$ determine the effect of the corresponding measurements at the update step of Kalman filter. A noisier measurement is considered to be less trustworthy. Increasing the corresponding element in $R_k$ will give less weight to a less trustworthy measurement during the Kalman filter update.

It is known that the wheel encoders provide accurate measures of linear and angular velocity when the robot has no motion. The measurements will be simply zero, while the gyro and camera estimation can be affected by sensor noise and environment conditions. However, when the robot is in motion, measurements from wheel encoders become less trustworthy. Equation (8) and (9) show that when linear and angular velocity measurements from wheel encoders are both zero, the robot is still, and $\sigma^\omega$, $\sigma^\rho$ are zero, which means the wheel encoder estimation dominates the other two sensors modalities. If the robot is in motion, the values of $\sigma^\omega$, $\sigma^\rho$ are proportional to velocity, therefore wheel encoders are trusted less, and measurements from gyro and vision-based estimation are taken into account to improve the estimates.

For comparison, a Kalman filter without visual odometry method is also designed. The state vector and matrices in (5) remain the same. There are only three measurements, namely the first three measurements in (7). Therefore, the measurement vector of the reduced system $y_k \in \mathbb{R}^3$ and the measurement matrix $H_k$ are made of first three rows of $H_k$. In the reduced system, both the wheel encoders and gyro measure only velocity states. The position states are updated by the state update operation in (5), and amount to an integration of velocity over time. Experimental results the two systems and their comparison are given in next section.

IV. EXPERIMENTAL RESULTS

Experiments were performed to test the proposed localization method. For comparison purposes, the full system fusing vision-based estimation, IMU measurements and wheel encoders is compared to the reduced system fusing only IMU and wheel encoders. The robot used in experiments is a Pioneer 3-DX differential drive robot. A laptop with Intel Core Duo processor was fixed to the robot collected all measurements and performed all signal processing and Kalman filter operations. The IMU is a Nintendo Wii Remote (Wiimote) with MotionPlus. This IMU costs approximately $42 (U.S. dollars), and represents an extremely low cost option for an IMU.

A. Testing the Reduced System

In order to establish the performance of the reduced system, simple experiments were performed. A human operator drove the robot along a simple path in the hallways of the Engineering and Computer Science building at the University of Texas at Dallas. The path begins facing east and makes four left turns of $90^\circ$ to return to the starting point. The total distance is approximately 100m.
Fig. 4 is a cropped section of an image that the reduced system renders while in operation. The background image is from Google maps, showing the location of the experiment. As seen in the legend of Fig. 4 the star denotes the starting point of the robot. The green dashed line was added afterwards in order to show the path that the robot was driven along. The red line represents the path estimated using only the wheel encoders to measure angular velocity. The blue “blob” represents the path of the robot estimated by using only the Wiimote’s measurement of angular velocity. The magenta line represents the path of the robot estimated using the reduced system’s Kalman filter-based fusion of both sets of angular velocity measurements.

The wheel encoders have several sources of error, which cause the path estimate to appear crooked. As the distance traveled increases, the encoders become less and less reliable. The MEMS rate-gyros inside the Wiimote’s have an extreme bias, which is proportional to the ambient air temperature of the environment. This causes the Wiimote’s estimated path to appear as circling around in an epicyclic manner. Inspection of Fig. 4 shows that the Wiimote estimate is useless on its own. Despite this, the reduced system manages to create an accurate and useful estimate of its path simply by fusing the devices’ faulty measurements and estimating the bias of the IMU. This gives a good example that the reduced system is accurate. The addition of pose and orientation estimates and an additional set of velocity measurements will ideally improve the estimate further.

B. Results of the Full System and Comparison With the Reduced System

The full system is described in the body of this paper. The experimental setup includes the same equipment described earlier. Fig. 5 shows the setup for the full system experiments and all of the needed measurement devices. The Wiimote and camera are mounted on top of the robot. The camera used is a Matrix Vision BlueFox with USB 2 interface. It has a resolution of 1024 × 768 pixels. The frame rate of the camera, when including all image processing and Kalman filter calculations, is 24 frames/second. The camera is calibrated to learn its intrinsic parameters and lens distortion parameters. In order to establish an accurate baseline across multiple experiments, two laser pointers are mounted on the robot facing the floor, one on the front and one on the back.

The camera is mounted such that the center of the camera’s imaging surface is aligned with the robot’s center of rotation. Efforts were made to align the camera axes with the robot axes. A target consisting two white rectangles provide a set of eight co-planar feature points. In this paper, the Wiimote measures only angular velocity and thus can be mounted anywhere on the robot as long as the axes are aligned and it is tightly strapped down to the surface of the robot.

A straight-forward experiment was devised and ran for each system. The robot is placed at the starting point as pictured in Fig. 6. At the start, the robot’s wheels are aligned with two marks on the start line. The laser pointers are used to make sure the robot always has the same start position, which is marked on the floor. For the full system, a human operator selects the eight planar feature points that are tracked with the Kanade-Lucas tracker [30].

The remainder of the experiment is fully automated and is the same for both systems. The robot is commanded to move forward 1500 millimeters, then stop and rotate −30°. Next, it is commanded to drive backwards 1500mm, stop, then forward 1500mm again. After that, it stops and rotates
Fig. 7. Typical estimation results of one trial: Pose V.S. time

Fig. 8. Typical estimation results of one trial: Velocity V.S. time

Fig. 9. Typical estimation results of one trial: position on y-axis V.S. position on x-axis

Fig. 10. Summary of Experimental Data (10 trials of each system)

+30°, then moves backward another 1500mm. Theoretically, the robot should be at its starting position, but errors will have caused some drift. The laser pointers are used to note the stopping point, which provides a ground truth. Before performing the experiments, we first measure the distances from the robot center of rotation (the mid-point between two wheels) to the front laser point (called a) and back laser point (called b) respectively. After each trial, the positions of laser points are recorded as $a'$ and $b'$. Therefore the new position of robot center can be located on line $a'b'$, and the offset to the starting point is measured. The orientation offset is calculated as

$$\theta = \arctan\left(\frac{y_{a'} - y_{b'}}{x_{a'} - x_{b'}}\right)$$

where $(x_{a'}, y_{a'})$ and $(x_{b'}, y_{b'})$ are the coordinates of $a'$ and $b'$ in the world frame.

Figs. 7-9 shows typical estimation results for a run of experiment. Due to the drift inherent to the wheel encoders, the robot never gets back to the exact starting point. The full system gives an accurate estimate of the robot state over time and the final robot pose.

Computation time did not allow both estimates to run at the same time. The experiment was repeated 10 times for each system and the error between the estimated position and the ground truth was recorded for each trial. The sample mean and sample standard deviation of the error along the world frame x-axis, y-axis, the Euclidean norm of the pose error, and the world frame orientation $\theta$, are all presented side by side for both systems in Table I and graphically in Fig. 10. From these results it is clear that both the mean and variance of full system are significantly lower than that of the reduced system for all measurements. The results show that the full system with visual odometry effectively improves the accuracy of the pose estimate when a planar scene is kept in the field of view.

V. CONCLUSION

This paper presents a novel sensor fusion approach for localization of a mobile robotic system. Continuous and discrete forms of the Euclidean Homography Matrix are
used to recover pose and velocity information from camera images. Wheel encoders and a low-cost IMU (a Wii Remote controller), measure the linear and angular velocity of the robot. A Kalman filter is used to fuse the velocity and pose estimates from different sensors. Experiments were performed to explore the performance of the proposed method. It can be seen from experimental results that neither the wheel encoders nor the Wiimote by themselves can be used alone to accurately localize a robot. Fusion of wheel odometry with an IMU does give a somewhat accurate estimate of the actual robot motion. Fusing this type of system with the proposed visual odometry method further improves the accuracy of the system with respect to localization.

There are several avenues open for future work. Further refinement of the presented system may improve performance further. Methods such as virtual parallax and/or the continuous and discrete Essential matrix will be used to perform visual odometry with non-coplanar feature points. A method must be determined for handling large scale movements of the robot and camera, such that feature points can be allowed to enter and leave the camera field of view without disrupting the estimation, such as in [31]. Finally, extensive experiments will be performed where the robot will navigate through large-scale environments.

TABLE I

<table>
<thead>
<tr>
<th>Axis</th>
<th>Full System</th>
<th>Reduced System</th>
</tr>
</thead>
<tbody>
<tr>
<td>(f_x)</td>
<td>-5.8271</td>
<td>12.9903</td>
</tr>
<tr>
<td>(f_y)</td>
<td>-7.3093</td>
<td>28.9184</td>
</tr>
<tr>
<td>(|\text{R}|)</td>
<td>9.3478</td>
<td>31.7021</td>
</tr>
<tr>
<td>(\theta_w)</td>
<td>1.0829</td>
<td>1.4260</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(\mu(E))</th>
<th>(\sigma(E))</th>
<th>(\mu(E))</th>
<th>(\sigma(E))</th>
</tr>
</thead>
<tbody>
<tr>
<td>-5.8271</td>
<td>12.9903</td>
<td>9.7607</td>
<td>15.7477</td>
</tr>
<tr>
<td>-7.3093</td>
<td>28.9184</td>
<td>-6.4058</td>
<td>53.0139</td>
</tr>
<tr>
<td>9.3478</td>
<td>31.7021</td>
<td>63.1645</td>
<td>55.3034</td>
</tr>
<tr>
<td>1.0829</td>
<td>1.4260</td>
<td>4.9971</td>
<td>3.4915</td>
</tr>
</tbody>
</table>

REFERENCES