Fusion of Fixation and Odometry for Vehicle Navigation

Amit Adam, *Student Member, IEEE,* Ehud Rivlin, *Member, IEEE,* and Héctor Rotstein, *Member, IEEE*

***Abstract*—This paper deals with the problem of determining the position and orientation of an autonomous guided vehicle (AGV) by fusing odometry with the information provided by a vision system. The main idea is to exploit the ability of pointing a camera in different directions, to fixate on a point of the environment while the AGV is moving. By fixating on a landmark, one can improve the navigation accuracy even if the scene coordinates of the landmark are unknown. This is a major improvement over previous methods which assume that the coordinates of the landmark are known, since any point of the observed scene can be selected as a landmark, and not just pre-measured points. This work argues that fixation is basically a simpler procedure than previously mentioned methods. The simplification comes from the fact that only one point needs to be tracked as opposed to multiple points in other methods. This disposes of the need to be able to identify which of the landmarks is currently being tracked, through a matching algorithm or by other means. We support our findings with both experimental and simulation results.**

***Index Terms*— Data fusion, fixation, landmark-based navigation.**

I. Introduction

TO be able to operate appropriately, an autonomous guided vehicle (AGV) must keep a sufficiently accurate track of its location and orientation with respect to some given coordinate system. Depending on the function of the AGV, this navigation task may be either relatively simple and easy to achieve, or quite complicated and critical. For instance, if the vehicle is operated in a controlled environment, with frequent interaction with an operator, then simple sensory information like odometry, can probably be used to achieve a satisfactory precision. On the other hand, if the vehicle is to operate autonomously for relatively long periods of time in a potentially unfriendly environment, then more sophisticated navigation solutions must be found. The different measurements available for navigation may be classified into two groups:

1) **Dead-Reckoning**. In this group, the AGV starts from a given known location, and then integrates the reading of its instruments in order to compute its local position. Examples of dead-reckoning systems are odometry and inertial navigation.
2) **External Measurements**. In this group, the AGV gets external information from the environment via additional sensors. The external information may be of a global nature, or it may be local information. The information may also be either complete or partial. As an example of complete global information, the AGV may be equipped with a GPS receiver and a map of the environment. If the AGV gets relative information of its location and orientation with respect to some reference points with unknown localization, then the external information is partial and local. An example of this type is given by the information received from a CCD.

Being self-contained, dead-reckoning has intrinsic advantages in terms of implementation simplicity, robustness, and large bandwidth. However, since dead-reckoning is based on direct instrument integration, inevitable inaccuracies, undermodeling, instrument errors and noise result in location and orientation errors which grow unboundedly with time. Application of dead-reckoning is then limited by the accuracy of the instruments and the duration of the autonomous navigation period. On the other hand, systems relying on external information are usually less straightforward to implement, less reliable, and provide information at a slower rate. However, although the instantaneous errors may be relatively large, these errors are bounded and do not grow with time. Consequently they can be used when only coarse but bounded location/orientation is required. It follows from this discussion that dead-reckoning and external measurement navigation have basically complementary features. It is then natural to attempt to combine or *fuse* the two strategies in order to obtain improved results.

Traditionally, AGV's have used their built-in odometry to compute their location and orientation using dead-reckoning. More recently, cameras mounted on a computer controlled variable pan-tilt platform have become "off-the-shelf" and can be mounted on an AGV for performing a variety of tasks, not necessarily connected with navigation. Motivated by this fact, this paper deals with the problem of determining the position of an AGV by fusing information from odometry with information coming from the vision system. In particular, the possibility of pointing the camera in different directions, is used to fixate on a point of the environment while the AGV is moving. As discussed below, the measurements of the angle and angular rate to the fixated point can then be used as external information.

Manuscript received June 19, 1998; revised July 11, 1999.

A. Adam is with the Department of Mathematics, Technion—Israel Institute of Technology, Haifa 32000, Israel (e-mail: amita@tx.technion.ac.il).

E. Rivlin is with the Department of Computer Science, Technion—Israel Institute of Technology, Haifa 32000, Israel (e-mail: ehudr@cs.technion.ac.il).

H. Rotstein is with Rafael—Armament Development Authority, Haifa 31021, Israel. He is also with the Department of Electrical Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel (e-mail: hector@ee.technion.ac.il).

Publisher Item Identifier S 1083-4427(99)08392-7.

Several authors have fused external information with odometry in order to obtain a better position estimate of an AGV. For instance, range sensor readings and a map of the operating environment were fused in [8] with odometry. By detecting laser reflections from reflectors placed in known locations, the angles from the vehicle to certain known points were computed and fused with odometry in [13] and [17]. In [7] odometry was fused with information coming from the detection of the position of a known landmark in an image taken of the vicinity of the landmark. In [5] the additional information was angle readings to three known landmarks.

The topic of self motion estimation from a sequence of images has a distinguished history in computer vision [1]. More specifically, the utilization of fixation for recovering egomotion information can be found in [10], [11], and [15]. Of particular interest for the present research is [14], where the relationship between the translation and rotation vectors describing the egomotion is discussed. The use of fixation for motion estimation is considered in [10], [11], and [15], where it is shown that fixation enhances the ability to recover the focus of expansion (FOE). As will be seen, measurement of the FOE is part of the fusion process discussed below.

References [6] and [12] discuss fixation and navigation. In [6], fixation is used as a control cue to guide a vehicle along a predetermined path. An initial range estimate to the fixation point is assumed. Localization through fixation is discussed in [12]. However range measurements which may be obtained using stereo are needed in their formulation.

In the quite different aeronautical and marine setup, fixation has also been considered under the name "bearings-only" measurements. In that context, the observer is assumed to be either located at a fixed position or moving with accurate information on self-motion. The observer then takes bearing, i.e., line-of-sight angles, measurements to a target and uses these measurements to track the location of the target. This sort of "dual" problem was considered in [2] and [4] and references therein. The basic approach in these works is to assume a simple stochastic model for the movements of the target, like constant target velocity plus normal white noise process, and use the measurements to recover the unknown motion parameters.

The purpose of this paper is to present a new way of fusing odometry with the measurements generated by a camera mounted on an AGV. The camera is provided with two degrees of mechanical freedom, allowing to keep the image of a certain fixed point at the same place on the screen over a period of time. Fixation is a relatively simple computational task which can be performed in real-time while the AGV is moving. Information coming from the camera system is then fused with odometry readings to obtain a more accurate estimate of the position and orientation of the vehicle.

The paper is divided into two parts. In the first part, it is shown that by fixating on a landmark one can improve the navigation accuracy even if the scene coordinates of the landmark are unknown. This is a major improvement over previous methods which assume that the coordinates of the landmark are known. In particular, using this approach any point of the observed scene can be selected as the landmark, and not just pre-measured points. In the second part of the paper, we assume we have an initial estimate of the coordinates of the fixation point. Then very simple measurements are needed from the vision system—only pan and tilt angles from the vehicle to the point. We formulate a fusion process of these measurements with odometry information, and describe experimental validation of this procedure. We also suggest an "emergency procedure" for obtaining absolute position once the vehicle gets lost. Thus, the second part shows that fixation is a simpler procedure than previously mentioned methods. The simplification comes from the fact that only one point needs to be tracked as opposed to multiple points in other methods [5], [7]. This disposes of the need to be able to identify which of the landmarks is currently being tracked, through a matching algorithm or by other means.

The framework used to fuse the fixation and odometry information is the Extended Kalman Filter (EKF) framework [3].

The next section describes the system being proposed and the constraints fixation imposes. Section III describes the EKF formulation of the ideas formulated in Section II. Section IV describes a formulation which uses simpler measurements from the vision system. Section V presents simulation and experimental results. Conclusions are presented in the last section.

II. System Description and Fixation Constraints

The system used for studying the fusion of odometry with fixation consists of an autonomous vehicle provided with a video camera. The AGV is equipped with an encoder on each one of the two motorized wheels, generating odometry measurements. The camera is mounted on a pan and tilt platform on top of the AGV. The camera fixates on a certain point as the vehicle moves, and is instrumented so that at all times the pan and tilt angles with respect to the vehicle can be measured. Three coordinate systems are relevant in this setup: a "scene" system which is assumed to be static, a "body" system attached to the AGV, and an "image" system attached to the camera. These coordinate systems are detailed next.

Consider first the scene coordinate system $(\hat{x}_s, \hat{y}_s, \hat{z}_s)$ such that the $\hat{z}_s$ axis is the vertical one, and the movement of the vehicle is constrained to lie in the $\hat{x}_s, \hat{y}_s$ plane. The position and orientation of the vehicle can then be specified by its coordinates (x_r, y_r) in the scene, and its heading angle θ_r. The two-dimensional (2-D) vector (x_r, y_r) specifies the position of the midpoint between the two motorized wheels, while θ_r is the angle between the x_r axis and the direction perpendicular to the baseline between the two wheels, measured in anti-clockwise direction. The body coordinate system can consequently be obtained from the scene coordinate system by a translation followed by a rotation around $\hat{z}_s$.

The attitude of the camera with respect to the body is specified (Fig. 1) by means of the pan and tilt angles θ_c and φ respectively. When looking straight ahead in the direction of motion $\theta_c = 0$; when looking to the left, $\theta_c = \pi/2$. The condition $\varphi > 0$ means that the camera is looking upwards.

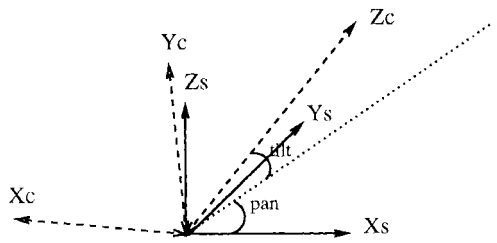

Fig. 1. Scene and camera coordinates.

Thus the configuration of the camera and vehicle is specified by $(x_r, y_r, \theta_r, \theta_c, \varphi)$.

Consider next a perspective projection model for the camera, with focal length equal to 1, and assume that the center of projection is located exactly above the midpoint between the encoded wheels, at a fixed height h. Define a camera coordinate system $(\hat{x}_c, \hat{y}_c, \hat{z}_c)$ as follows: $\hat{z}_c$ is the camera viewing direction—i.e. the optical axis, $\hat{y}_c$ is the vertical direction on the image screen, and $\hat{x}_c$ is chosen to form a right handed system. The origin of the camera coordinate system is at the point specified by $(x_r, y_r, h)^t$ in scene coordinates. With these definitions the transformation between scene coordinates and camera coordinates is given by

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = R\left[\begin{pmatrix} x_s \\ y_s \\ z_s \end{pmatrix} - \begin{pmatrix} x_r \\ y_r \\ h \end{pmatrix}\right] \tag{1}$$

where

$$R = \begin{pmatrix} -\sin\theta & \cos\theta & 0 \\ -\sin\varphi\cos\theta & -\sin\varphi\sin\theta & \cos\varphi \\ \cos\varphi\cos\theta & \cos\varphi\sin\theta & \sin\varphi \end{pmatrix} \tag{2}$$

and $\theta = \theta_r + \theta_c$.

Assume now the camera and vehicle at time t are in the configuration

$$\chi(t) = (x_r(t), y_r(t), \theta_r(t), \theta_c(t), \varphi(t)).$$

As the AGV moves in the environment, the camera coordinates of a fixed scene point change with time. Let $\vec{p} = (x(t), y(t), z(t))^t$ be the camera coordinates of the scene point $(x, y, z)^t$ when the camera is in configuration $\chi(t)$, as given by (1). Assuming that the movement is smooth enough to allow differentiation, the derivative of $\vec{p}$ can be computed. The computation (which is omitted for reasons of space) leads to the following expression for the derivative:

$$\begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{pmatrix} = -R(t)\begin{pmatrix} \dot{x}_r(t) \\ \dot{y}_r(t) \\ 0 \end{pmatrix} + \begin{pmatrix} \dot{\varphi}(t) \\ -\dot{\theta}(t)\cos\varphi(t) \\ -\dot{\theta}(t)\sin\varphi(t) \end{pmatrix} \times \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} \tag{3}$$

where $R(t)$ is the matrix R in (2) evaluated with $\theta_r(t)$, $\theta_c(t)$ and $\varphi(t)$. Thus we have expressed $\dot{p}$ as

$$\dot{p} = -\vec{t} - \omega\hat{k} \times \vec{p}. \tag{4}$$

From standard kinematics (see, for example, [9]) we know that from this representation we can infer that the instantaneous motion of the camera is a rotation and a translation. The rotation is around the vector

$$\hat{k} = \frac{1}{\sqrt{\dot{\varphi}^2 + \dot{\theta}^2}}\begin{pmatrix} -\dot{\varphi}(t) \\ \dot{\theta}(t)\cos\varphi(t) \\ \dot{\theta}(t)\sin\varphi(t) \end{pmatrix} \tag{5}$$

at an angular rate of $\omega = \sqrt{\dot{\varphi}^2 + \dot{\theta}^2}$, while the translation is with a velocity vector

$$\vec{t} = R(t)\begin{pmatrix} \dot{x}_r(t) \\ \dot{y}_r(t) \\ 0 \end{pmatrix} = \begin{pmatrix} -\dot{x}_r\sin\theta + \dot{y}_r\cos\theta \\ -\dot{x}_r\sin\varphi\cos\theta - \dot{y}_r\sin\varphi\sin\theta \\ \dot{x}_r\cos\varphi\cos\theta + \dot{y}_r\cos\varphi\sin\theta \end{pmatrix}. \tag{6}$$

The vectors $\hat{k}, \vec{t}$ are both given in the camera coordinate system at time t. As described in the next two subsections, the fact that the fixated camera motion depends on the motion of the AGV can be exploited to generate useful navigation information.

A. Geometric Constraint

Suppose that the camera is fixated on a point with coordinates $\vec{R}_0 = (0, 0, z)^t$ in the camera coordinate system. If the motion normal to the line-of-sight is nonzero, then the vectors $\omega\hat{k}$ and $\vec{t}$ describing the motion satisfy the constraint [14]

$$\omega\hat{k} = \omega_{R_0}\hat{R}_0 + \frac{1}{||R_0||}\vec{t} \times \hat{R}_0 \tag{7}$$

where ω_{R_0} is some (unknown) constant. More explicitly

$$\begin{pmatrix} -\dot{\varphi}(t) \\ \dot{\theta}(t)\cos\varphi(t) \\ \dot{\theta}(t)\sin\varphi(t) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \omega_{R_0} \end{pmatrix} + \frac{1}{z}\begin{pmatrix} -\dot{x}_r\sin\theta + \dot{y}_r\cos\theta \\ \sin\varphi(-\dot{x}_r\cos\theta - \dot{y}_r\sin\theta) \\ \cos\varphi(\dot{x}_r\cos\theta + \dot{y}_r\sin\theta) \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \tag{8}$$

After performing the cross product one obtains

$$\begin{pmatrix} -\dot{\varphi}(t) \\ \dot{\theta}(t)\cos\varphi(t) \\ \dot{\theta}(t)\sin\varphi(t) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \omega_{R_0} \end{pmatrix} + \frac{1}{z}\begin{pmatrix} -\dot{x}_r\sin\varphi\cos\theta - \dot{y}_r\sin\varphi\sin\theta \\ \dot{x}_r\sin\theta - \dot{y}_r\cos\theta \\ 0 \end{pmatrix}. \tag{9}$$

In order to eliminate the unknown distance z to the fixation point, one can now divide (on both sides of the equation) the y coordinate by the x coordinate and obtain

$$\frac{\dot{\theta}\cos\varphi}{-\dot{\varphi}} = \frac{\dot{x}_r\sin\theta - \dot{y}_r\cos\theta}{-\dot{x}_r\sin\varphi\cos\theta - \dot{y}_r\sin\varphi\sin\theta} \tag{10}$$

or after some further manipulation

$$\begin{aligned} &g_1(\dot{x}_r, \dot{y}_r, \dot{\theta}_r, \theta_r, \theta_c, \varphi, \dot{\theta}_c, \dot{\varphi}) \\ &\quad = \dot{\varphi}(\dot{x}_r\sin\theta - \dot{y}_r\cos\theta) \\ &\qquad - \dot{\theta}\sin\varphi\cos\varphi(\dot{x}_r\cos\theta + \dot{y}_r\sin\theta) = 0. \end{aligned} \tag{11}$$

This equation means that given camera motion measurements $(\theta_c, \varphi, \dot{\theta}_c, \dot{\varphi})$ and assuming the camera was fixated, the motion

parameters of the vehicle $(\dot{x}_r, \dot{y}_r, \dot{\theta}_r)$ are constrained by (11). As shown later, the angle $\dot{\theta}_r$ actually does not appear in this equation.

B. Computer Vision Constraint

The constraint (11) is a geometric constraint arising from the fact that the line of sight passes through a fixed point in space as the vehicle moves. A related condition can be obtained via computer vision considerations. Suppose that two or three images taken by the camera are available during the motion. It is standard [11], [15] that, under certain assumptions, a point (U, V) called the "focus of expansion" (FOE) can be computed from the collection of images. The translation vector describing the motion of the camera is related to the FOE via the equation

$$\vec{t} = \lambda \begin{pmatrix} U \\ V \\ 1 \end{pmatrix}. \tag{12}$$

In other words, the direction of the translation can be obtained from the FOE. By using (6) we get

$$\vec{t} = \lambda \begin{pmatrix} U \\ V \\ 1 \end{pmatrix} = \begin{pmatrix} -\dot{x}_r \sin\theta + \dot{y}_r \cos\theta \\ -\dot{x}_r \sin\varphi \cos\theta - \dot{y}_r \sin\varphi \sin\theta \\ \dot{x}_r \cos\varphi \cos\theta + \dot{y}_r \cos\varphi \sin\theta \end{pmatrix}. \tag{13}$$

Again we may now divide the y coordinate by the x coordinate on both sides of the equation to get

$$\frac{V}{U} = \frac{-\sin\varphi(\dot{x}_r \cos\theta + \dot{y}_r \sin\theta)}{-(\dot{x}_r \sin\theta - \dot{y}_r \cos\theta)} \tag{14}$$

or

$$g_2(\dot{x}_r, \dot{y}_r, \dot{\theta}_r, \theta_r, \theta_c, \varphi, U, V) = 0. \tag{15}$$

Although the coordinates (U, V) of the FOE are not available directly, they can be estimated from the sequence of images and consequently one can assume that the camera plus FOE estimator act as a "soft sensor," providing the FOE coordinates as pseudo-measurements. Therefore, for the case under consideration, the camera measurements are $(\theta_c, \varphi, U, V)$ which are to be compared with $(\theta_c, \varphi, \dot{\theta}_c, \dot{\varphi})$ when the geometric constraint is being used.

At this point it is worth emphasizing that there is a significant difference between the geometric constraint and the computer vision constraint. The geometric constraint states that the quantities $\dot{x}_r$, $\dot{y}_r$ describing the platform movement and the quantities θ, φ, $\dot{\theta}$, $\dot{\varphi}$ describing the camera, are related due to camera fixation. From this relationship one can attempt to update the platform movement estimate, based on the camera movement. The computer vision constraint states the same thing in principle; but in this case physical or geometrical considerations enter only indirectly. Indeed, the computer vision constraint is an expression of the fact that the soft computer vision sensor can also estimate the platform translation (through the estimation of the camera translation), and this estimate can be used in conjunction with the odometry $(\dot{x}_r, \dot{y}_r)$ estimate to get a better overall estimate of platform movement. Thus the computer vision constraint is intuitively expected to be less useful than the geometrical constraint. Indeed, in the experiments we did we saw that a nearly ideal soft sensor is required for the computer vision formulation to work well.

III. EKF Formulation

The geometrical and computer vision constraints described above can now be fused with odometry measurements, in order to reduce navigation errors. The fusion is performed using the extended Kalman filter described next.

The first step is to express the quantities $(\dot{x}_r, \dot{y}_r, \dot{\theta}_r)$ as functions of the odometry readings. Let r_1, r_2 be the radii of the right and left wheels respectively, and let $\Delta\varphi_1, \Delta\varphi_2$ be the outputs of the right and left encoders. By simple kinematic arguments [16], when the right and left wheels traverse distances $Q_1 = r_1\Delta\varphi_1$ and $Q_2 = r_2\Delta\varphi_2$, respectively, the vehicle moves a distance $\Delta s = (Q_1 + Q_2)/2$ and its orientation changes by $\Delta\theta_r = (Q_1 - Q_2)/B$. Here B denotes the baseline distance between the two wheels. The change in scene coordinates is correspondingly

$$\Delta x = \Delta s \frac{\sin(\Delta\theta_r/2)}{\Delta\theta_r/2} \cos(\theta_r + \Delta\theta_r/2)$$
$$\Delta y = \Delta s \frac{\sin(\Delta\theta_r/2)}{\Delta\theta_r/2} \sin(\theta_r + \Delta\theta_r/2).$$

If the encoder sampling rate is fast enough, then the angle θ_r can be assumed to be constant between samples. The term $\sin(\Delta\theta_r/2)/(\Delta\theta_r/2)$ can then be ignored so that

$$\dot{x}_r \approx (\Delta s/\Delta t)\cos(\theta_r + \Delta\theta_r/2) \tag{16}$$
$$\dot{y}_r \approx (\Delta s/\Delta t)\sin(\theta_r + \Delta\theta_r/2) \tag{17}$$
$$\dot{\theta}_r = \Delta\theta_r/\Delta t. \tag{18}$$

Note that (16)–(18) give $(\dot{x}_r, \dot{y}_r, \dot{\theta}_r)$ as functions of the odometry readings Q_1, Q_2 and the current orientation θ_r. Further examination of the constraints (11), (15) show that these equations involve the scalar products of the following three vectors

1) $\vec{v} = (\dot{x}_r, \dot{y}_r)$ which is the heading direction of the vehicle;
2) $\vec{l} = (\cos(\theta_r + \theta_c), \sin(\theta_r + \theta_c))$ which is a vector along the line of sight to the fixation point;
3) $\vec{n} = (-\sin(\theta_r + \theta_c), \cos(\theta_r + \theta_c))$ which is the normal to $\vec{l}$.

Cancelling the angle θ_r from these scalar products, and plugging (16)–(18) into (11) and (15) yields

$$g_1(Q_1, Q_2, \theta_c, \varphi, \dot{\theta}_c, \dot{\varphi}) = D(\dot{\varphi}\sin a - h\sin\varphi\cos\varphi\cos a) \tag{19}$$

$$g_2(Q_1, Q_2, \theta_c, \varphi, U, V) = D(U\sin\varphi\cos a - V\sin a) \tag{20}$$

where

$$D = D(Q_1, Q_2) = \frac{Q_1 + Q_2}{2\Delta t}$$
$$a = a(\theta_c, Q_1, Q_2) = \theta_c - \frac{Q_1 - Q_2}{2B}$$
$$h = h(\dot{\theta}_c, Q_1, Q_2) = \dot{\theta}_c + \frac{Q_1 - Q_2}{B\Delta t}.$$

In the face of the equations above, consider the state vector formed by the two odometry variables

$$\vec{x} = \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix}.$$

The vector of camera measurements is

$$\vec{c} = \begin{pmatrix} \theta_c \\ \varphi \\ \dot{\theta}_c \\ \dot{\varphi} \end{pmatrix} \quad \text{or} \quad \vec{c} = \begin{pmatrix} \theta_c \\ \varphi \\ U \\ V \end{pmatrix}$$

depending on the vision system and constraint being used.

In order to describe the EKF algorithm, suppose that in the last time interval the vector $\hat{x}_0$ of current estimate of the odometry values has been computed, with an associated covariance matrix P_0. Let $\hat{Q}_1$, $\hat{Q}_2$ denote the odometry measurements, and let $\hat{c}$ be the vector of current camera measurements. The camera measurements are assumed to be corrupted by white Gaussian noise, with associated covariance matrix P_{cam}. The proposed value for the propagated state vector is

$$\hat{x}^- = \lambda \hat{x}_0 + (1-\lambda)\begin{pmatrix} \hat{Q}_1 \\ \hat{Q}_2 \end{pmatrix} = F(\hat{x}_0, \hat{Q}_1, \hat{Q}_2)$$

where $0 \leq \lambda \leq 1$. The motivation for this estimate is the following. If the odometry were error-free, then $(\hat{Q}_1, \hat{Q}_2)^t$, i.e., $\lambda = 0$, would give the correct propagated value for the state. In the face of inevitable odometry errors, it is possible to exploit the fact that the path followed by an AGV is such that true odometry values in subsequent time intervals are strongly correlated. One can consequently use the previous estimate of the state to reduce the odometry noise in $(\hat{Q}_1, \hat{Q}_2)^t$.

Note that the parameter λ is used as a simple method to smooth some of the fluctuations in the odometry readings. We emphasize that even with $\lambda = 0$ our filter works in a satisfactory manner. A possible improvement of the prediction phase is to take into account the path the vehicle is supposed to traverse and make λ vary accordingly: in segments where the path is without acceleration λ should be high. An inner loop Kalman filter may be employed here with the prediction of the state being based on the trajectory the vehicle should perform at the current time step, and the update of the state being based on the odometry readings. We thank a reviewer for this comment.

Assuming that the odometry measurement noise is uncorrelated with the current odometry error, a straightforward manipulation shows that the covariance of the predicted estimate $\hat{x}^-$ is

$$\begin{aligned} P_1^- &= \frac{\partial F}{\partial \hat{x}_0} P_0 \left(\frac{\partial F}{\partial \hat{x}_0}\right)^t + \frac{\partial F}{\partial \hat{Q}_1}\sigma_1^2 \left(\frac{\partial F}{\partial \hat{Q}_1}\right)^t \\ &\quad + \frac{\partial F}{\partial \hat{Q}_2}\sigma_2^2 \left(\frac{\partial F}{\partial \hat{Q}_2}\right)^t \\ &= \lambda^2 P_0 + (1-\lambda)^2 \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix} \end{aligned}$$

where σ_i^2 is the variance of $\hat{Q}_i$ around Q_i. Next, update the prediction $\hat{x}^-$ using the camera measurement $\hat{c}$ as follows. Choose g as g_1 or g_2 from (19) or (20) and compute

$$H_1 = \frac{\partial g}{\partial \vec{x}} = \left(\frac{\partial g}{\partial Q_1}\frac{\partial g}{\partial Q_2}\right)\Big|_{(\hat{x}^-, \hat{c})}$$

$$H_2 = \frac{\partial g}{\partial \vec{c}} = \left(\frac{\partial g}{\partial \theta_c}\frac{\partial g}{\partial \varphi}\frac{\partial g}{\partial(\dot{\theta}_c \text{ or } U)}\frac{\partial g}{\partial(\dot{\varphi} \text{ or } V)}\right)\Big|_{(\hat{x}^-, \hat{c})}.$$

The Kalman gain is computed as

$$K = P_1^- H_1^t \left(H_1 P_1^- H_1^t + H_2 P_{\text{cam}} H_2^t\right)^{-1}$$

and the update equations are

$$\begin{aligned} \hat{x} &= \hat{x}^- + K(-g(\hat{x}^-, \hat{c})) \\ P_1 &= (I - KH_1)P_1^-. \end{aligned}$$

(Note that since the measurements (11) and (15) are implicit, we use the implicit Kalman filter formulation).

Given the new estimate $\hat{x}$ for (Q_1, Q_2) of the current time interval, compute the new position and orientation of the vehicle by adding the quantities in (16)–(18) multiplied by Δt to the current values of (x_r, y_r, θ_r). On doing this, the quantities Δs and $\Delta\theta_r$ are computed by using the values of $\hat{x}$.

IV. Using Angles Only Measurements

In the previous sections, it was shown how fixation may be used to update the estimates of the quantities Q_1, Q_2 which determine the motion of the vehicle. These quantities can subsequently be integrated to obtain the position of the vehicle. The measurements needed to carry out that procedure included the camera angles, and in addition either the derivatives of these angles or the FOE. This section shows that if the measurements obtained from the vision system are restricted to the camera angles only, fixation can still be used as an aid to odometry. Notice that in the present case an initial estimate of the coordinates of the fixation point must be available. It is also important to stress that the angle measurements and the coordinates of the point are not used to compute the position of the vehicle directly. To do that, additional information in the form of a range measurement or angles to another landmark would be required. Instead, the angle measurements and the best known estimate of the relative position of the point with respect to the vehicle are used to estimate the true dynamics of the vehicle, based on the approximate dynamics reported by odometry.

Suppose that the camera fixates on a point with scene coordinates $(a, b, c)^t$. Let $x = a - x_r$, $y = b - y_r$ and $z = c - h$ be the relative position of the point with respect to the camera, and let $\vec{m}$ be the change in odometry reading between time $k-1$ and k. Due to unavoidable errors, when computing the movement of the vehicle based on these measurements, the difference between the current and the computed location will increase with time. The evolution of the error $\vec{b}_\theta$ between the real displacement and the one reported by the odometry is illustrated in Fig. 2. If the odometry reports a displacement vector $\vec{m}_0$ along the x axis, then the real displacement can be written as $\vec{r} = \vec{m}_0 + \vec{b}_0$. Consequently, when the

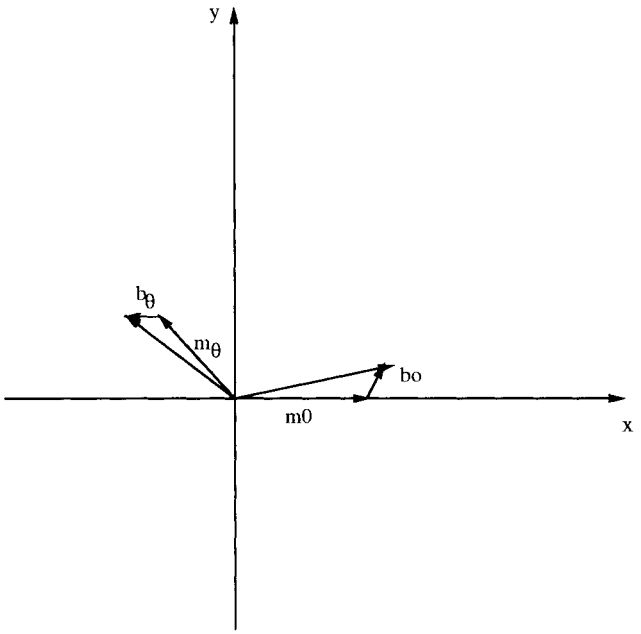

Fig. 2. Odometry bias model.

odometry reading reports $\vec{m}$ in direction $\theta = \arg\vec{m}$, the real displacement is given by $\vec{m} + \vec{b}_\theta$, where

$$\vec{b}_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \vec{b}_0. \tag{21}$$

The odometry errors can now be incorporated as additional states to the state-vector

$$\vec{x} = \begin{pmatrix} x \\ y \\ z \\ \vec{b}_0 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ b_{0x} \\ b_{0y} \end{pmatrix} \tag{22}$$

The dynamics associated with this state-vectors can be written as follows. If from time $k-1$ to k the odometry reports a displacement $\vec{m}$, then letting $\theta = \arg\vec{m}$, an improved estimate for the state at time k is

$$\begin{pmatrix} \hat{x}_k^- \\ \hat{y}_k^- \end{pmatrix} = \begin{pmatrix} \hat{x}_{k-1} \\ \hat{y}_{k-1} \end{pmatrix} - (\vec{m} + \vec{b}_\theta)$$
$$\hat{z}_k^- = \hat{z}_{k-1}$$
$$\hat{b}_{0k}^- = \hat{b}_{0k-1}$$

where

$$\vec{b}_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \hat{b}_{0k-1}. \tag{23}$$

Thus, before incorporating the measurements, the propagated state-vector is given by

$$\hat{x}_k^- = A_\theta x_{k-1} + B\vec{m} \tag{24}$$

where

$$A_\theta \doteq \begin{pmatrix} 1 & 0 & 0 & -\cos\theta & \sin\theta \\ 0 & 1 & 0 & -\sin\theta & -\cos\theta \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{25}$$

and

$$B \doteq \begin{pmatrix} -1 & 0 \\ 0 & -1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}. \tag{26}$$

Using the standard EKF equations, if P_{k-1} denotes the covariance of the state at time $k-1$, the covariance propagated to time k is given by

$$P_k^- = A_\theta P_{k-1} A_\theta^t$$

The next step is to take the measurement $\hat{c} = (\hat{\theta},\ \hat{\varphi})^t$. Consider

$$g(\vec{x}) = \begin{pmatrix} g_1 \\ g_2 \end{pmatrix} = \begin{pmatrix} \arctan\frac{y}{x} \\ \arctan\frac{z}{\sqrt{x^2+y^2}} \end{pmatrix} \tag{27}$$

and define

$$H = \left.\frac{\partial g}{\partial \vec{x}}\right|_{(\hat{x}_k^-)}. \tag{28}$$

Similarly as before, the Kalman gain is computed as

$$K = P_k^- H^t (H P_k^- H^t + P_{\text{cam}})^{-1}$$

and the update equations are

$$\hat{x}_k = \hat{x}_k^- + K(\hat{c} - g(\hat{x}_k^-))$$
$$P_k = (I - KH)P_k^-.$$

A. Localization Procedure

The formulation above is suitable for the case when the vehicle starts at a known location, and the camera measurements suffice to keep the odometry errors bounded. If, alternatively, the initial location is unknown, fixation has not been achieved for some time, or errors have grown so large that the EKF does not longer perform satisfactorily, then an alternative approach is required. In this section, a localization procedure based on fixation is presented. The procedure is to be invoked when the absolute position error is large, i.e. the vehicle is "lost."

During the procedure, a certain path (say a square) is followed by the vehicle, while the camera is fixating on a fixation point. Assume the odometry displacement reports are accurate with respect to the large absolute position error. Define the state vector to be $\vec{x} = (x, y, z)^t$, i.e., only the relative position coordinates. By following the same Kalman filter formulation as above, the position will get updated as the vehicle follows the path. Once the updates stabilize and conform with the odometry reported displacements, the vehicle is localized once again. The difference from the previous scenario is that in the first scenario the initial state is assumed to be close to the true state (in the position coordinates). In this scenario, however, the initial state is very distant from the true state.

One must note however that for this localization procedure to work, the vehicle should know its orientation in the scene coordinate system: we assume the measurement vector is $\hat{c} = (\hat{\theta},\ \hat{\varphi})^t$ [see (27)] and θ is the pan angle with respect to the scene x axis. In addition, since we are estimating the

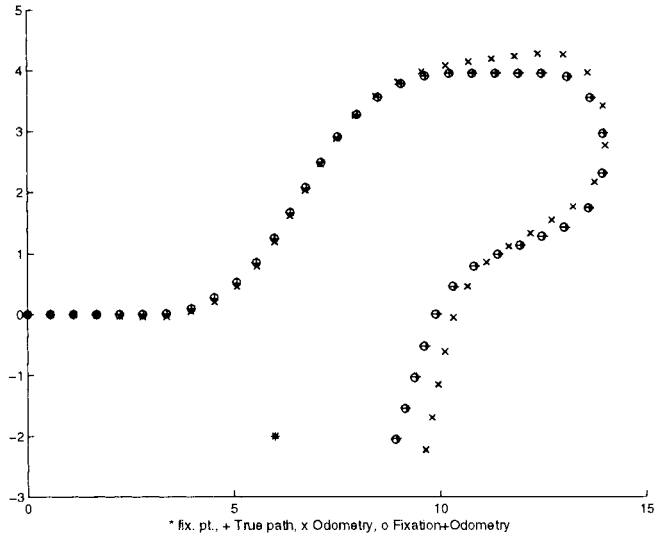

Fig. 3. Sample run—graphical results.

relative position with respect to the landmark, in order to know our absolute position we must of course know the absolute coordinates of the landmark.

V. Results

A. Simulation Results

The geometric constraint algorithm was tested by running a simulation program. The program "moves" the "true" vehicle according to some preset path. At each time step the "true" odometry readings are corrupted by adding a normal, zero-mean distributed noise, with variance proportional to the squared "true" value. In addition the "true" camera readings are corrupted with noise. These "true" readings may be simulated since we know the vehicle's "true" position. The corrupted odometry readings and camera readings are fed into the algorithm, and an estimated path is computed and displayed.

The simulated vehicle is moving at a speed of about 70 cm/s. The radii of the wheels is 5 cm and the baseline distance is 40 cm. Odometry noise was assumed to be normal with zero mean and a standard deviation of about 3% of the readings value. Later on a bias was also simulated by adding to one of wheels' true odometry value a uniform random variable between $(-1/3, 2/3)$ multiplied by some constant, so that again the standard deviation will be around 3% of the true value. The camera measurement noise was assumed to be normal with zero mean and standard deviation as follows:

- 1° for the (θ_c, φ) angle readings;
- $3\sigma = 10\%$ of the true value for $(\dot{\theta}_c, \dot{\varphi})$ readings.

The initial standard deviations of the errors in the odometry readings were taken as 3% of the true value. The λ parameter was taken as 0.5.

Fig. 3 shows graphically the results of a sample run. The figure shows the "true" path followed by the vehicle, the path that was computed based solely on odometry readings, and the path computed using fusion of camera measurements and odometry readings. We ran the simulation for 100 times and recorded for every run the average (along the path) odometry error and the average error in the position estimate based on our method. Fig. 4 compares these errors. The contribution of fixation to the reduction of odometry errors may be seen.

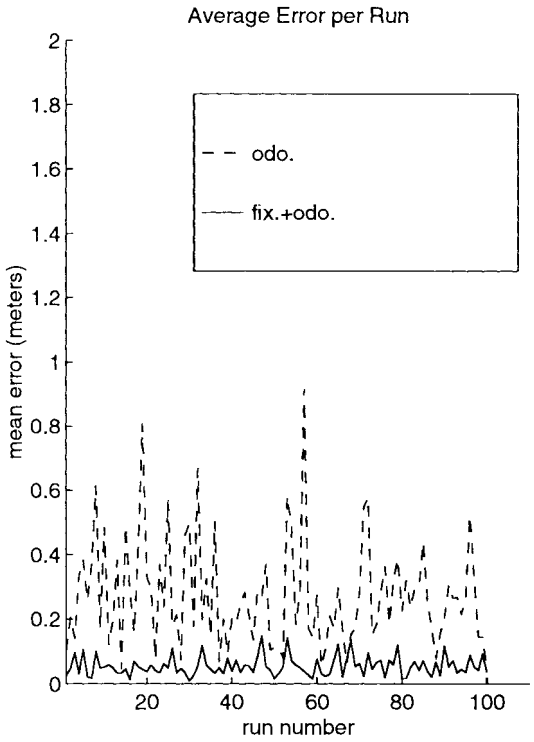

Fig. 4. Per run average odometry versus average fixation error.

We have checked consistency of the Kalman filter by using the measures described in [4], namely NEES, NIS, and whiteness of innovations. NEES is the normalized estimation error squared and should be distributed Chi square with 2 DOF (our state vector is 2-D). It is used to check that the variability in the actual errors of the estimated state indeed agrees with the covariance matrix that the filter estimates. Similarly NIS which is the normalized innovation squared should be Chi square with 1 DOF. The whiteness test checks correlation between innovations at different times. We have checked whiteness between time step k and time step $k+1$ (for every k).

We have summed the NEES values over the 100 runs, for every time step. This sum should be Chi square with 200 DOF which may be approximated as normal by the Central Limit Theorem. Fig. 5 shows this sum after normalization to a standard (zero mean, unit variance) normal variable. The range this normalized sum has to be in with probability of 95% is marked on the figure. As we can see we cannot reject the hypothesis that the NEES is distributed as expected. Similarly, Fig. 6 shows the normalized sum of 100 NIS values for every time step, and the range in which this sum should be with the same probability of 95%.

Correlation between innovations at subsequent time steps was checked both by looking at the 100-run sample autocorrelation, and by looking at the time-average autocorrelation of each of the 100 runs by itself. Figs. 7 and 8 show the

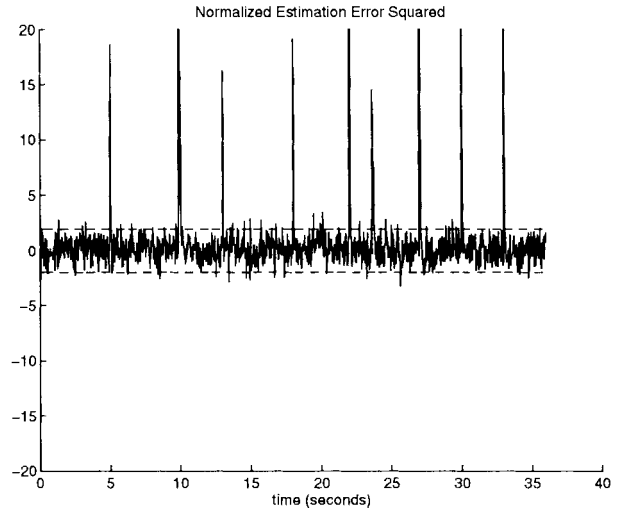

Fig. 5. NEES statistics.

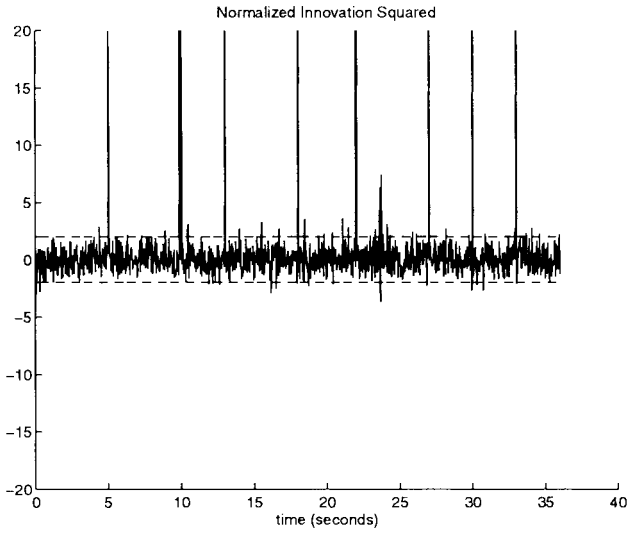

Fig. 6. NIS statistics.

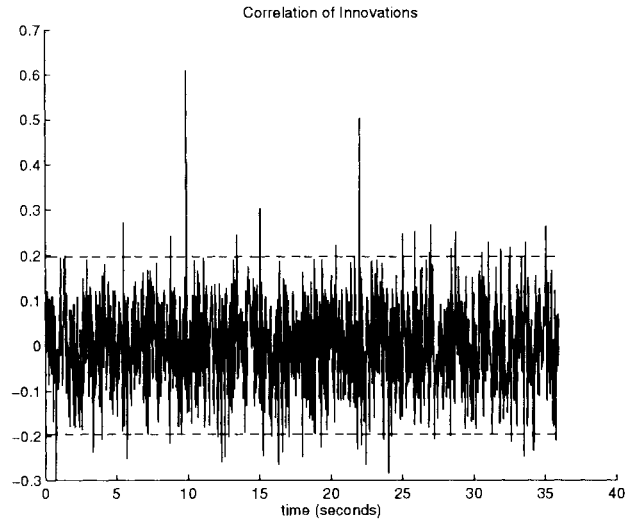

Fig. 7. Whiteness test—100-run sample autocorrelation.

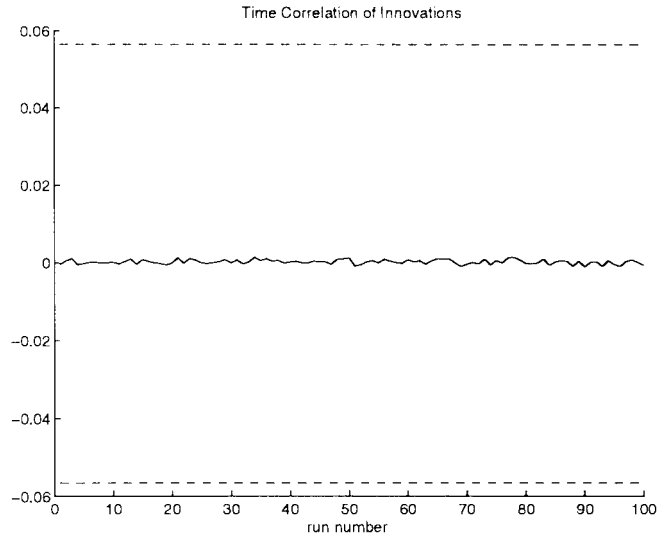

Fig. 8. Whiteness test—time average autocorrelation.

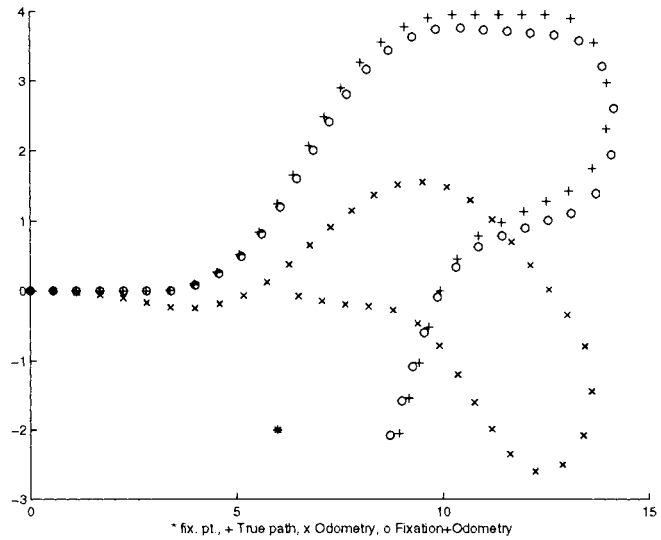

Fig. 9. Biased noise—graphical results.

computed statistics and the range in which they should fall with probability 95%. Once again it is evident we cannot reject the hypothesis that the innovations form a white noise process.

B. Biased Noise

Fig. 9 presents graphically the results of a sample run where odometry readings were corrupted with biased noise. The readings of the left wheel were biased upwards and indeed it is evident in the figure that the path computed from odometry readings tends to turn to the right. It can be seen that when the camera measurements are fused with those readings, the error is reduced. Fig. 10 shows the average errors along the path for 100 runs of biased odometry noise.

C. Experimental Results

Two experiments were conducted to test the formulation presented in Section IV, using a Nomad 200 robot

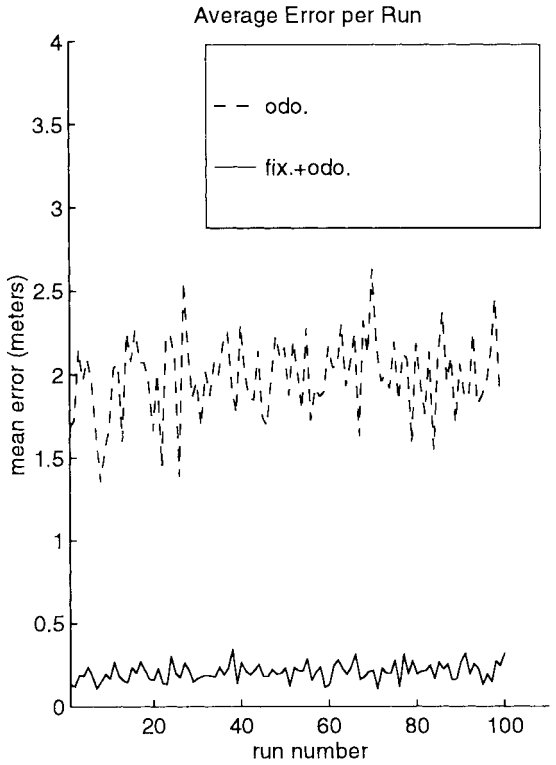

Fig. 10. Odometry versus fixation—biased noise.

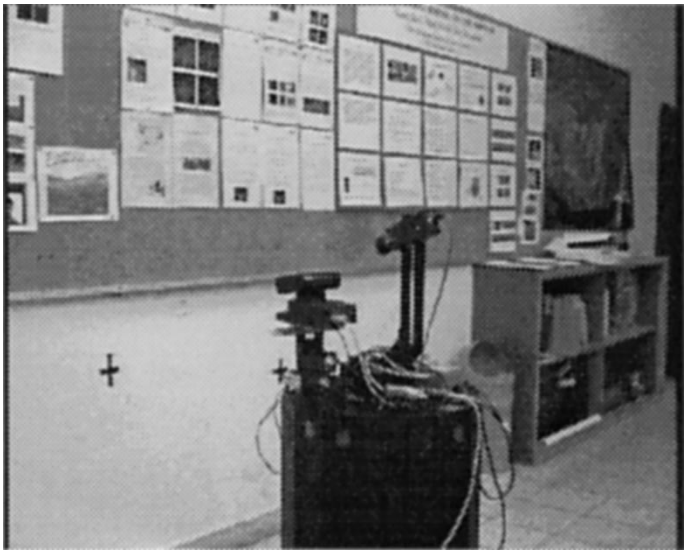

Fig. 11. Instrumented Nomad 200.

instrumented with a Canon VC-C1 video camera, as shown in Fig. 11.

In the first experiment the robot moved back and forth along a straight line segment 3 m long. Every 50 cm the camera fixated on the fixation point, which was about 2 m away. The segment was traversed six times. Fig. 12 shows in a dashed line the odometry errors, and in a continuous line the errors in the position determined by the EKF formulated in Section IV. The average error along the x axis is 4.2 cm. as compared to a 15.4 cm average odometry error, and along the y axis the

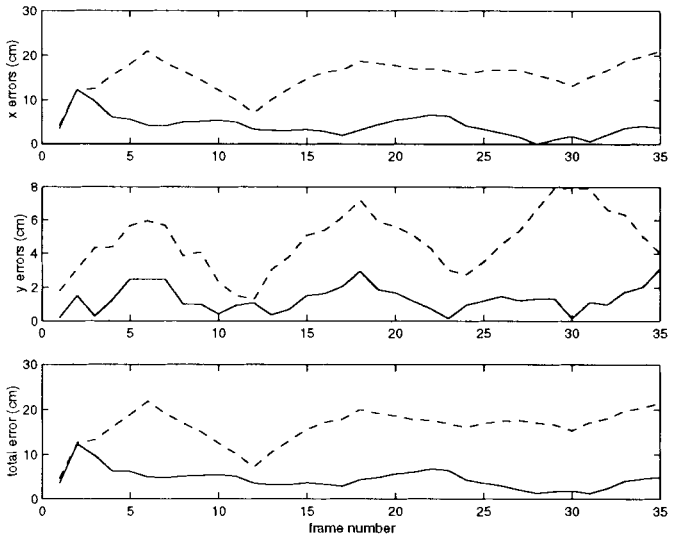

Fig. 12. Experimental odometry versus fixation errors.

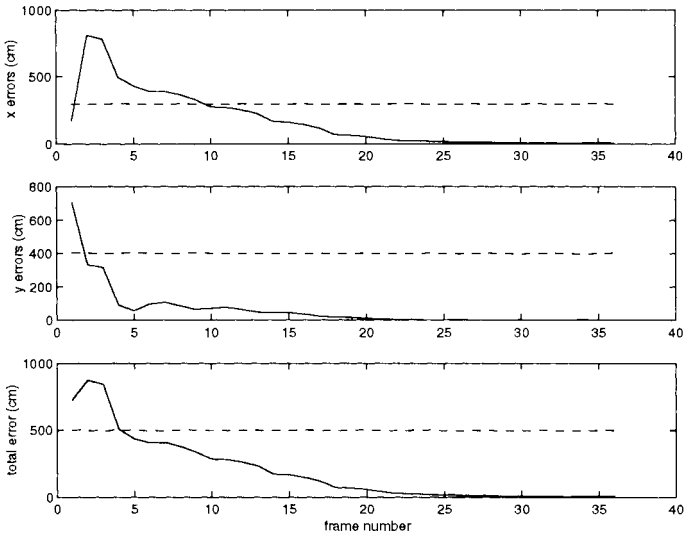

Fig. 13. Localization experiment—initial location: $(-300, -400)$.

average error is 1.3 cm as compared with a 4.8 cm average odometry error.

The second experiment shows the feasibility of the localization procedure suggested previously. The robot was placed at $(0,0)$. The odometry counters were set to a point which is a few meters away—thus simulating a scenario where the robot is lost. The robot then moved along a square path while fixating on a point. The length of the side of the square was 60 cm. At each corner of the square the camera angles were read. The absolute position was then estimated using an iterated EKF as described in the previous section. Figs. 13–15 show how the error in absolute position decreases as more and more measurements are taken. The final error is a few centimeters although the initial error was a few meters.

We remark that a regular EKF did not suffice for the localization procedure. In the first two experiments (Figs. 13 and 14) five iterations of the EKF were sufficient, and in the third experiment (Fig. 15) 25 iterations were used.

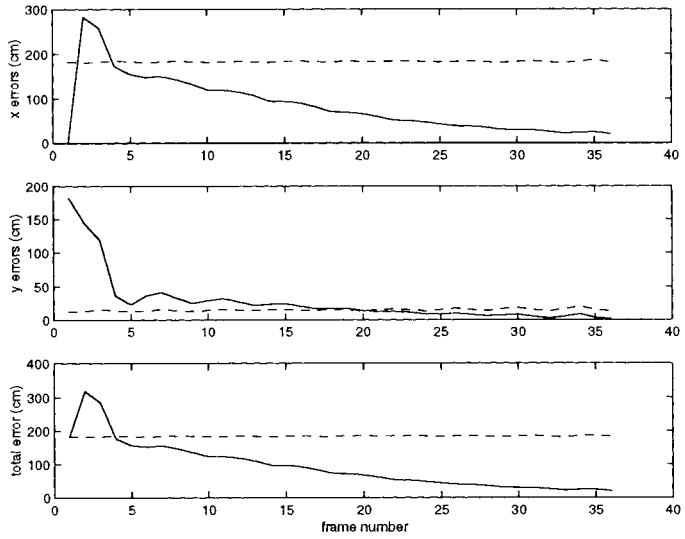

Fig. 14. Localization experiment—initial location: $(-185, 15)$.

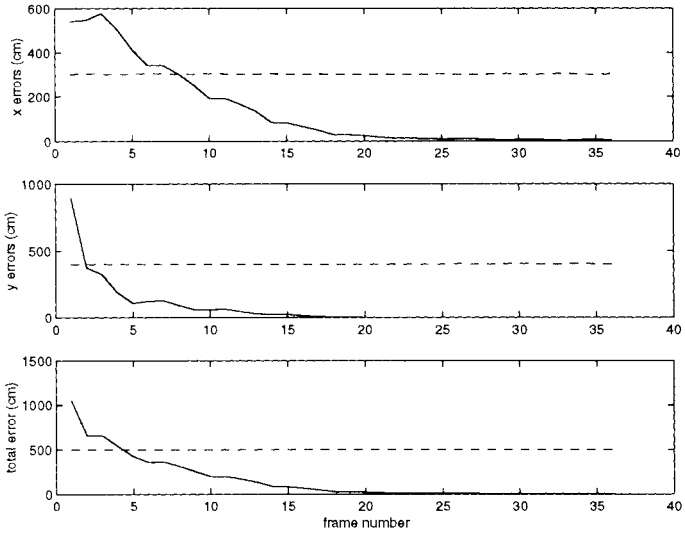

Fig. 15. Localization experiment—initial location: $(300\,400)$.

VI. Conclusions and Further Work

This paper addressed the problem of autonomous guided vehicle navigation. Exploiting the technical possibility to instrument the vehicle with a point-able camera, the approach presented is to use odometry as the basic navigation tool, but to reduce the inevitable odometry errors by an external measurement generated by the camera mounted on the AGV. The camera is performing fixation on a certain point. Since fixation is a relatively simple computational task, the camera may be able to fixate on the point continuously as the vehicle is moving. Thus the algorithm does not require the vehicle to stop every once in a while for computing its position. Since the fusion of odometry and camera measurements is done by a Kalman filter and is not computationally expensive, the only cost of the algorithm is a dedicated camera.

It was shown that by fixating on a landmark one can improve the navigation accuracy even if the scene coordinates of the landmark are unknown. This is a major improvement over previous methods which assume that the coordinates of the landmark are known. In particular, using this approach any point of the observed scene may be selected as the landmark, and not just pre-measured points.

It was also shown that if an initial estimate of the coordinates of the landmark does exist, then very simple measurements are needed from the vision system. This was verified experimentally. Since only one point needs to be tracked by the vision system, there is no need to be able to distinguish between different landmarks as is done in other works. This makes the implementation of the algorithm a simpler task.

Further work is needed to analyze the effects of choosing different fixation points. The performance of the algorithm varies as the geometry of the fixation point and the vehicle's path change. Observability analysis of the system is expected to give some insight on the optimal choice of a fixation point given a certain path the vehicle should traverse.

References

[1] J. K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images—A review," *Proc. IEEE*, vol. 76, pp. 917–935, Aug. 1988.

[2] V. J. Aidala, "Kalman filter behavior in bearings-only tracking applications," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-15, pp. 29–39, Jan. 1979.

[3] N. Ayache and O. D. Faugeras, "Maintaining representations of the environment of a mobile robot," *IEEE Trans. Robot. Automat.*, vol. 5, pp. 804–819, Dec. 1989.

[4] Y. Bar-Shalom and X.-R. Li, *Estimation and Tracking: Principles, Techniques and Software*. Norwood, MA: Artech House, 1993.

[5] Ph. Bonnifait and G. Garcia, "A multisensor localization algorithm for mobile robots and its real-time experimental validation," in *Proc. IEEE Int. Conf. Robotics Automation*, 1996, pp. 1395–1400.

[6] R. N. Braithwaite and B. Bhanu, "Robust guidance of a conventionally steered vehicle using destination bearing," in *Proc. 1993 Int. Symp. Intelligent Control*, 1993, pp. 382–387.

[7] F. Chenavier and J. L. Crowley, "Position estimation for a mobile robot using vision and odometry," in *Proc. IEEE Int. Conf. Robotics Automation*, 1992, pp. 2588–2593.

[8] I. J. Cox, "Blanche—An experiment in guidance and navigation of an autonomous robot vehicle," *IEEE Trans. Robot. Automat.*, vol. 7, pp. 193–204, Apr. 1991.

[9] J. J. Craig, *Introduction to Robotics*, 2nd ed. Reading, MA: Addison-Wesley, 1989.

[10] K. Daniilidis, "Fixation simplifies 3d motion estimation," *Comput. Vis. Image Understand.*, vol. 68, pp. 158–169, Nov. 1997.

[11] C. Fermuller and Y. Aloimonos, "The role of fixation in visual motion analysis," *Int. J. Comput. Vis.*, vol. 11, pp. 165–186, Oct. 1993.

[12] D. W. Murray, I. D. Reid, and A. J. Davison, "Steering and navigation behaviors using fixation," in *Brit. Mach. Vis. Conf.*, 1996.

[13] T. Nishizawa, A. Ohya, and S. Yuta, "An implementation of onboard position estimation for a mobile robot," in *Proc. IEEE Int. Conf. Robotics Automation*, 1995, pp. 395–400.

[14] M. A. Taalebinezhaad, "Direct recovery of motion and shape in the general case by fixation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 847–853, Aug. 1992.

[15] S. Toelg, "On the finite kinematics of visual fixation," Tech. Rep. CAR-TR-736, Univ. Maryland, Sept. 1994.

[16] C. M. Wang, "Location estimation and uncertainty analysis for mobile robots," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 1988, pp. 1230–1235.

[17] Y. Watanabe and S. Yuta, "Position estimation of mobile robots with internal and external sensors using uncertainty evolution technique," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 1990, pp. 2011–2016.

Amit Adam (S'99) received the B.Sc. degree (summa cum laude) from Ben-Gurion University, Israel, in 1989, and the M.Sc. degree from the Weizmann Institute of Science, Rehovot, Israel, in 1996, both in mathematics. He is currently pursuing the Ph.D. degree at the Intelligent Systems Laboratory, Computer Science Department, Technion—Israel Institute of Technology, Haifa.

From 1989 to 1996, he worked in operations research for the Israeli Air Force. His areas of interest are computer vision and robot navigation.

Ehud Rivlin (S'90–M'95) received the B.Sc. and M.Sc. degrees in computer science and the M.B.A. degree from the Hebrew University, Jerusalem, Israel. He received the Ph.D. degree from the University of Maryland, College Park.

He is currently Associate Professor, Computer Science Department, Technion—Israel Institute of Technology, Haifa. His current research interests are in machine vision and robot navigation.

Hector Rotstein (M'94) received the Ingeniero Electricista degree from the Universidad Nacional del Sur, Argentina, in 1985 and the M.Sc. and Ph.D. degrees in electrical engineering from the California Institute of Technology, Pasadena, in 1990 and 1993, respectively.

From 1985 to 1989, he was an Engineer with the Control Group, Planta Piloto de Ingenieria Quimica, Bahia Blanca, Argentina. From 1993 to 1997, he was with the Department of Electrical Engineering, Technion—Israel Institute of Technology, Haifa. Since 1997 he has been with Rafael—Armament Development Authority, Haifa, as a Research Engineer. His current research interests include robust optimal control, the control of active vision systems, and the application of system theory to model autonomous systems.