Hindawi Publishing Corporation
Journal of Robotics
Volume 2011, Article ID 548042, 12 pages
doi:10.1155/2011/548042

Research Article

Monocular Vision-Based Robot Localization and Target Tracking

Bing-Fei Wu, Wang-Chuan Lu, and Cheng-Lung Jen

Department of Electrical Engineering, National Chiao Tung University, 1001 University Road, Hsinchu 300, Taiwan

Correspondence should be addressed to Cheng-Lung Jen, cljen0130@gmail.com

Received 14 June 2011; Revised 15 October 2011; Accepted 16 October 2011

Academic Editor: Huosheng Hu

This paper presents a vision-based technology for localizing targets in 3D environment. It is achieved by the combination of different types of sensors including optical wheel encoders, an electrical compass, and visual observations with a single camera. Based on the robot motion model and image sequences, extended Kalman filter is applied to estimate target locations and the robot pose simultaneously. The proposed localization system is applicable in practice because it is not necessary to have the initializing setting regarding starting the system from artificial landmarks of known size. The technique is especially suitable for navigation and target tracing for an indoor robot and has a high potential extension to surveillance and monitoring for Unmanned Aerial Vehicles with aerial odometry sensors. The experimental results present "cm" level accuracy of the localization of the targets in indoor environment under a high-speed robot movement.

1. Introduction

Knowledge about the environment is a critical issue for autonomous vehicle operations. The capability of localizing targets with a robot for environment is highly demanded. In [1], it investigates the vision-based object recognition technique for detecting targets of the environment which have to be reached by a robot. After the robot finishes the target detection, it is an essential task for robots to know the positions of targets. The tasks include navigation and object tracking. As a result, the approach to localize targets for specific tasks is an essential issue in some applications. For UAVs, there are some similar demands [2, 3] that UAVs have to track the positions of ground objects for reconnaissance or rescue assistance with monocular vision. Besides, UAVs need to observe the changing surroundings to understand the movements of the aircraft better, or the localizing system has to direct the aircraft to a region of interest after taking ground observations.

In recent years, odometry sensors have been widely used for estimating the motion of vehicles moving in a 3D space environment such as UAVs and Unmanned Ground Vehicles (UGVs). For instance, Inertial Navigation Systems (INSs) are applied to measure linear acceleration and rotational velocity, and capable of tracking the position, velocity and attitude of a vehicle by integrating these signals [4] or mobile vehicles use the Global Position System (GPS) and Inertial Measurement Units (IMUs) for land vehicle applications [5]. However, most of inertial navigation system sensors are expensive for some applications in the indoor environment. Optical wheel encoders and an electrical compass provide linear and angular velocities respectively. Both of two sensors are basic odometry sensors and widely used owing to their low cost, simplicity, and easy maintenance. Encoders provide a way of measuring the velocity to estimate the position of the robot and compasses are often used to detect the orientations of the robot. Based on the sensor information, motion control is done and then the localization of the robot is estimated [6, 7]. Despite some limitations of an encoder, most researchers agree that an encoder is a vital part of the robot's navigation system and the tasks will be simplified if the encoder accuracy is improved. Besides, the additional sensor, a camera, allows a robot to perform a variety of tasks autonomously. The use of computer vision for localization has been investigated for several decades. The camera has not been at the center of robot localization while most of researchers have more attention to other sensors such as laser range-finders and sonar. However, it is surprising that vision is still an attractive choice for sensors because cameras are compact, cheaper, well understood, and ubiquitous. In this paper, the algorithm is achieved by combining different types

of sensors including optical wheel encoders, an electrical compass, and a single camera.

Mobile vehicles such as robots, UGVs, and UAVs are becoming fully intelligent robotic systems which can handle more complicated tasks and environments. It is necessary to provide them with higher autonomy and better functionalities. For instance, robot localization and target tracking are mandatory. For these tasks, some of sensors such as GPSs, IMUs, laser range-finders, encoders, and compasses are possibly combined to describe the time evolution of the motion model. Additionally, visual sensors, such as electro-optic or infrared cameras, have been included in the on-boarded sensor suit of many robots to increase their value. Therefore, it is a popular research topic regarding how to combine hybrid sensors to achieve a special task. Recent researches have shown that the way to improve sequential estimation and achieve repeatable motion with multisensors is to adopt the probabilistic method. The Kalman filter has been widely used for data fusion such as navigation systems [8], virtual environment tracking systems [9], and 3D scene modeling [10]. The Kalmna filter is a mathematical method to provide an iterated procedure for a least-square estimation of a linear system. The procedure includes predicting state by using the motion model and then correcting the state by the measurement innovation. The EKF is a varied type of the Kalman filter for a nonlinear system such as the target tracking system in this paper. More details about the Kalman filter are introduced in the material of [11–14]. In this paper, the algorithm represents a new approach of EKF-based multisensor data fusion. The information from different types of sensors is combined to improve the estimations in the system state. In order to make our system able to deal with the variety of tasks such as navigation for a mobile robot or ground target tracking for UAVs, we are determined to choose monocular vision rather than binocular vision as one type of our sensors. Based on the monocular vision, simultaneous robot localization and target tracking become more complex with a higher computational loading due to unknown depths of targets in only one image observation. Thus, it is an important issue regarding how to make use of computational tricks or adopt more efficient algorithms to reduce the running time. In this paper, Chelesky decomposition [15] and forward-and-back substitution are used to invert the covariance matrix for improving the computational efficiency because the property of covariance matrix is semipositive definite.

The rest of this paper is structured as follows. Section 2 presents the proposed algorithm and explains the details. Section 3 explains two key points about the proposed algorithm. The experimental results are provided in Section 4, and finally, Section 5 concludes the paper and suggests future work.

2. Explanation of the Algorithm

The algorithm is mainly divided into five parts including motion modeling, new target adding, measurement modeling, image matching, and EKF updating. By sequential measurement from sensors, the EKF is capable of improving the initial estimate for the unknown information while simultaneously updating the localization of targets and the robot pose. Finally, Chelesky decomposition and forward-and-back substitution are presented to calculate the inverse covariance matrix efficiently.

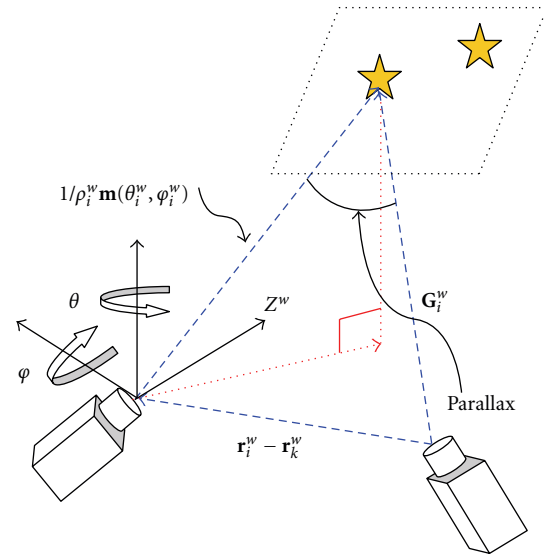

Figure 1: The robot moves from $\mathbf{r}_i^w$ to $\mathbf{r}_k^w$. Based on the two views, the parallax is produced. The initial depth can be refined to approach the real distance and the position of the ith target is estimated.

2.1. The Origins of the Proposed Algorithm. A single camera is mounted on our system as one of the sensors. The monocular vision infers that the depth of the target is not able to be measured by only one image but estimated by the sequential images. Therefore, the single camera has to estimate the depth by observing the target repeatedly to get parallax between different captured rays from the target to the robot. The orientations of target are estimated in the world coordinate system only by one image. $\mathbf{Y}_i^w$ is a six-dimension state vector and used to describe the position of the ith target in 3D space. Its equation is addressed as

$$\mathbf{Y}_i^w = \left[\mathbf{r}_i^{wT} \quad \theta_i^w \quad \varphi_i^w \quad \rho_i^w\right]^T. \tag{1}$$

$\mathbf{r}_i^w$ is the location of the robot when the robot detects the ith target in the first time. $\mathbf{G}_i^c$ is defined as the position of the target with respect to the camera coordinate system and denoted by

$$\mathbf{G}_i^c = \mathbf{R}^{cw}\left(\left(\mathbf{r}_i^w - \mathbf{r}_k^w\right) + \frac{1}{\rho_i^w}\mathbf{m}(\theta_i^w, \varphi_i^w)\right). \tag{2}$$

θ_i^w and φ_i^w are the orientations of the ith target and calculated by the pixel location of the target in only one image. The relationship between the position of the target $\mathbf{G}_i^c$ and the current robot localization $\mathbf{r}_k^w$ is presented in Figure 1.

$\mathbf{G}_i^w$ is the position of the target with respect to the world frame system. $1/\rho_i^w$ is defined as the distance from $\mathbf{r}_i^w$ to

the target and $\mathbf{m}(\theta_i^w, \varphi_i^w)$ is its unit vector. Consequently, $1/\rho_i^w \mathbf{m}(\theta_i^w, \varphi_i^w)$ is seen as the vector from $\mathbf{r}_i^w$ to the target and then $\mathbf{G}_i^w$ is calculated as $((\mathbf{r}_i^w - \mathbf{r}_k^w) + 1/\rho_i^w \mathbf{m}(\theta_i^w, \varphi_i^w))$. Finally, $\mathbf{G}_i^c$ is estimated if the depth $1/\rho_i^w$ is known in advance. However, the depth of the target is not able to be measured by a single image. The EKF is applied to estimate $1/\rho_i^w$, θ_i^w, and φ_i^w and they converge toward more correct values under the recursive iteration by using sequential image measurement information. To sum up, (2) is the basic concept and origin of our proposed algorithm.

2.2. Motion Modeling. We first derive the continuous-time system model that describes the time evolution in the state estimate. This model allows us to employ the sampled measurements of the wheel encoder and the compass for state propagation. The process state is described by the vector

$$\mathbf{Xm}_k = \begin{bmatrix} \mathbf{r}_k^{wT} & \mathbf{q}_k^{wT} & \mathbf{v}_k^{wT} & \boldsymbol{\omega}_k^{cT} \end{bmatrix}^T, \tag{3}$$

where $\mathbf{r}_k^w$ is the position of the robot with respect to the world frame. $\mathbf{q}_k^w$ is the quaternion that represents the orientation of the world frame for the robot. The linear velocity in the world frame and the angular velocity with respect to the camera frame are denoted by $\mathbf{v}_k^w$ and $\boldsymbol{\omega}_k^c$, respectively. In this system, the control inputs are measured by wheel encoders and the compass which provide the linear velocity and the angular velocity, respectively. Besides, the linear velocity is used to simulate the acceleration data of the robot for describing how components of the system state vector change completely. The definition of acceleration is defined as $\mathbf{a}_k^w = (\mathbf{v}_k^w - \mathbf{v}_{k-1}^w)/\Delta t$. In order to have similar on-line simulation, we define $\mathbf{a}_k^w$ as $(\mathbf{v}_k^w - \mathbf{v}_{k-1}^w)/\Delta t$ rather than $(\mathbf{v}_{k+1}^w - \mathbf{v}_k^w)/\Delta t$. Although the former definition is not better than the later one which is closer to the real acceleration, an iterated EKF is still capable of compensating for errors when adopting $(\mathbf{v}_k^w - \mathbf{v}_{k-1}^w)/\Delta t$ as the robot's acceleration. The system model describing the time evolution of the wheel encoders and of the compass is given by

$$\mathbf{Xm}_{k+1}^- = \begin{bmatrix} \mathbf{r}_{k+1}^w \\ \mathbf{q}_{k+1}^w \\ \mathbf{v}_{k+1}^w \\ \boldsymbol{\omega}_{k+1}^c \end{bmatrix} = \begin{bmatrix} \mathbf{r}_k^w + \mathbf{v}_k^w \Delta t + \frac{1}{2}\mathbf{a}_k^w \Delta t^2 \\ \mathbf{q}_k^w \times \mathbf{q}\left(\boldsymbol{\omega}_k^c + \boldsymbol{\Omega}_k^c\right) \\ \mathbf{v}_k^w + \mathbf{a}_k^w \Delta t \\ \boldsymbol{\omega}_k^c + \boldsymbol{\Omega}_k^c \end{bmatrix}. \tag{4}$$

The compass is used to measure $\Delta\theta_k$ which is defined as $(\theta_k - \theta_{k-1})$. $\boldsymbol{\omega}_{k+1}^c$ and $\boldsymbol{\omega}_k^c$ are defined as $(\theta_k - \theta_{k-1})/\Delta t$ and $(\theta_{k-1} - \theta_{k-2})/\Delta t$, respectively. If $\boldsymbol{\omega}_{k+1}^c$ is assumed to be $(\boldsymbol{\omega}_k^c + \boldsymbol{\Omega}_k^c)$, $\boldsymbol{\Omega}_k^c c$ is derived as $(\theta_k - 2\theta_{k-1} + \theta_{k-2})/\Delta t$. $\mathbf{q}_{k+1}^w$ is denoted by

$$\mathbf{q}_{k+1}^w = \begin{bmatrix} q_1 & -q_2 & -q_3 & -q_4 \\ q_2 & q_1 & -q_4 & q_3 \\ q_3 & q_4 & q_1 & -q_2 \\ q_4 & -q_3 & q_2 & q_1 \end{bmatrix} \begin{bmatrix} q_1' \\ q_2' \\ q_3' \\ q_4' \end{bmatrix}, \tag{5}$$

where

$$\mathbf{q}\left(\boldsymbol{\omega}_k^c + \boldsymbol{\Omega}_k^c\right) = \begin{bmatrix} q_1' \\ q_2' \\ q_3' \\ q_4' \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\Delta\theta_k}{2}\right) \\ u_x \sin\left(\frac{\Delta\theta_k}{2}\right) \\ u_y \sin\left(\frac{\Delta\theta_k}{2}\right) \\ u_z \sin\left(\frac{\Delta\theta_k}{2}\right) \end{bmatrix}. \tag{6}$$

$\mathbf{u} = \begin{bmatrix} u_x & u_y & u_z \end{bmatrix}^T$ is equal to $\boldsymbol{\omega}_{k+1}^c \Delta t / \|\boldsymbol{\omega}_{k+1}^c\| \Delta t$.

The covariance is predicted and modified if considering the control covariance $\mathbf{Cov}(\mathbf{v}_k^w, \boldsymbol{\omega}_k^c, \mathbf{a}_k^w)$. We assume that the process noise of the control vector is not correlated with the process state vector $\mathbf{Xm}_k$ so that $\mathbf{Cov}(\mathbf{Xm}_k, \mathbf{v}_k^w, \boldsymbol{\omega}_k^c, \mathbf{a}_k^w)$ is set to be a zero matrix. By using (A.4) in Appendix A, the predicted covariance of the system model is expressed as

$$\mathbf{Cov}\left(\mathbf{Xm}_{k+1}^-\right) = \mathbf{F}_{Xm}\mathbf{Cov}(\mathbf{Xm}_k)\mathbf{F}_{Xm}{}^T + \mathbf{Q}, \tag{7}$$

where

$$\mathbf{Q} = \mathbf{F}_v \mathbf{Cov}\left(\mathbf{v}_k^w\right)\mathbf{F}_v{}^T + \mathbf{F}_\omega \mathbf{Cov}\left(\boldsymbol{\omega}_k^c\right)\mathbf{F}_\omega{}^T + \mathbf{F}_a \mathbf{Cov}\left(\mathbf{a}_k^w\right)\mathbf{F}_a{}^T. \tag{8}$$

The first item $\mathbf{F}_{Xm}\mathbf{Cov}(\mathbf{Xm}_k)\mathbf{F}_{Xm}{}^T$ represents the noise from the process state vector and $\mathbf{Q}$ describes the noise from the measurement of the wheel encoders and the compass. The Jacobian matrix of $\mathbf{F}_{Xm}$ is shown as

$$\mathbf{F}_{Xm} = \frac{\partial \mathbf{f}}{\partial \mathbf{Xm}} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial \mathbf{r}_k^w} & \frac{\partial \mathbf{f}}{\partial \mathbf{q}_k^w} & \frac{\partial \mathbf{f}}{\partial \mathbf{v}_k^w} & \frac{\partial \mathbf{f}}{\partial \boldsymbol{\omega}_k^c} \end{bmatrix}_{13 \times 13}. \tag{9}$$

The Jacobian matrixes of $\mathbf{F}_v$ and $\mathbf{F}_\omega$ are as the same as $\partial \mathbf{f}/\partial \mathbf{v}_k^w$ and $\partial \mathbf{f}/\partial \boldsymbol{\omega}_k^c$, respectively.

2.3. Adding New Target. It is remarkable about our algorithm that new targets are initialized by using only one image. The initialization includes the state initial values and the covariance assignment.

2.3.1. Target Initialization. Equation (1) is used to define the location of the new target and estimate the six parameters of $\mathbf{Y}_i^w$ with the EKF prediction-update loop. If the robot senses the 1st target at the state k for the first time, the new target information is added and then the process state vector is modified. The expanded process state vector is represented by the following equation:

$$\mathbf{X}_k^- = \begin{bmatrix} \mathbf{Xm}_k^- \\ \mathbf{Y}_1^w \end{bmatrix}, \tag{10}$$

where

$$\mathbf{Y}_1^w = \begin{bmatrix} \mathbf{r}_1^{wT} & \theta_1^w & \varphi_1^w & \rho_1^w \end{bmatrix}_{6 \times 1}^T. \tag{11}$$

The first time observation of the 1st target is done at the current camera location $\mathbf{r}_1^w$. $\mathbf{m}(\theta_1^w, \varphi_1^w)$ is defined as an unit

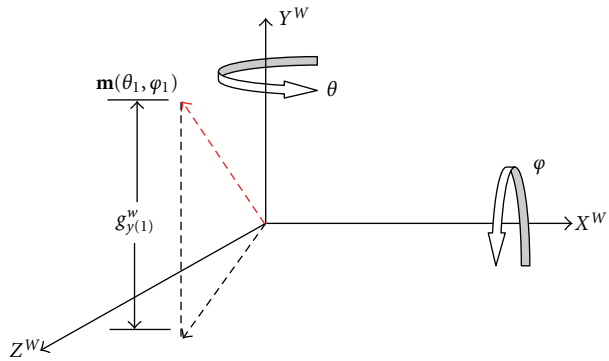

FIGURE 2: The orientations of the target with respect to the world frame can be estimated as a function of its undistorted pixel location $\mathbf{h}_u$.

vector from the location $\mathbf{r}_1^w$ to the 1st target with respect to the world frame. Figure 2 illustrates the relationship about $\mathbf{m}(\theta_1^w, \varphi_1^w)$, θ_1^w, and φ_1^w. $\mathbf{h}_u = [u\ v]^T$ is defined as the pixel location of the 1st target in an undistorted image. $\mathbf{g^c}$ is the location of the 1st target with respect to the camera frame. The location of the 1st target with respect to the world frame $\mathbf{g^w}$ is addressed as

$$\mathbf{g^w} = \begin{bmatrix} g_x^w \\ g_y^w \\ g_z^w \end{bmatrix} = \mathbf{R^{wc}}\mathbf{g^c} = \mathbf{R^{wc}} \begin{bmatrix} \dfrac{(u_0 - u)D_x}{f} \\ \dfrac{(v_0 - v)D_y}{f} \\ 1 \end{bmatrix} g_z^c, \tag{12}$$

where

$$\mathbf{R^{wc}} = \begin{bmatrix} q_1^2+q_2^2-q_3^2-q_4^2 & 2(q_2q_3 - q_1q_4) & 2(q_2q_4 + q_1q_3) \\ 2(q_2q_3 + q_1q_4) & q_1^2-q_2^2+q_3^2-q_4^2 & 2(q_3q_4 - q_1q_2) \\ 2(q_2q_4 - q_1q_3) & 2(q_3q_4 + q_1q_2) & q_1^2-q_2^2-q_3^2+q_4^2 \end{bmatrix}. \tag{13}$$

Then we get

$$\frac{\begin{bmatrix} g_x^w \\ g_y^w \\ g_z^w \end{bmatrix}}{g_z^c} = \mathbf{R^{wc}} \begin{bmatrix} \dfrac{(u_0 - u)D_x}{f} \\ \dfrac{(v_0 - v)D_y}{f} \\ 1 \end{bmatrix}. \tag{14}$$

Only g_x^w/g_z^c and g_y^w/g_z^c rather than g_x^w and g_y^w are computed by using $\mathbf{h}_u$. It is impossible to know g_z^c by only one image. However, it is possible to make use of g_x^w/g_z^c and g_y^w/g_z^c to calculate θ_1^w and φ_1^w which are shown as

$$\begin{bmatrix} \theta_1^w \\ \varphi_1^w \end{bmatrix} = \begin{bmatrix} \tan^{-1}\left(\dfrac{g_x^w}{g_z^c}, \dfrac{g_z^w}{g_z^c}\right) \\ \tan^{-1}\left(\dfrac{-g_y^w}{g_z^c}, \sqrt{\left(\dfrac{g_x^w}{g_z^c}\right)^2 + \left(\dfrac{g_z^w}{g_z^c}\right)^2}\right) \end{bmatrix}. \tag{15}$$

The unit vector $\mathbf{m}(\theta_1^w, \varphi_1^w)$ is derived as a function of the orientations of the target and described as

$$\mathbf{m}(\theta_1^w, \varphi_1^w) = \begin{bmatrix} \cos\varphi_1^w \sin\theta_1^w \\ -\sin\varphi_1^w \\ \cos\varphi_1^w \cos\theta_1^w \end{bmatrix}. \tag{16}$$

The final parameter $1/\rho_1^w$ is not able to be measured by only one image but estimated by the sequential images with EKF prediction-update loop. As a result, it is assumed to be an initial value $1/\rho_0$.

2.3.2. New Covariance Assignment. The system covariance is modified after adding a new target. By using (B.1) in Appendix B, the new covariance $\mathbf{Cov}(\mathbf{X}_k^-)$ and the function of $\mathbf{Y}_1^w$ are denoted by

$$\mathbf{Cov}\left(\mathbf{X}_k^-\right) = \begin{bmatrix} \mathbf{Cov}(\mathbf{X}\mathbf{m}_k) & \mathbf{B}_1^T \\ \mathbf{B}_1 & \mathbf{A}_1 \end{bmatrix},$$
$$\mathbf{Y_1^w} = \mathbf{k}\left(\mathbf{r}_k^w, \mathbf{q}_k^w, \mathbf{h}_d(u, v),\ \rho_1^w\right), \tag{17}$$

where

$$\mathbf{k} = \begin{bmatrix} \mathbf{r}_1^w \\ \theta_1^w \\ \varphi_1^w \\ \rho_1^w \end{bmatrix} = \begin{bmatrix} r_x \\ r_y \\ r_z \\ \tan^{-1}\left(\dfrac{g_x^w}{g_z^c}, \dfrac{g_z^w}{g_z^c}\right) \\ \tan^{-1}\left(\dfrac{-g_y^w}{g_z^c}, \sqrt{\left(\dfrac{g_x^w}{g_z^c}\right)^2 + \left(\dfrac{g_z^w}{g_z^c}\right)^2}\right) \\ \rho_1^w \end{bmatrix}. \tag{18}$$

$\mathbf{B}_1$ is not a zero matrix because the new target's location is correlated with the estimates of the robot. According to (B.4) and (B.5) in Appendix B, $\mathbf{B}_1$ and $\mathbf{A}_1$ are derived as

$$\mathbf{B}_1 = \mathbf{k}_x\mathbf{Cov}(\mathbf{X}\mathbf{m}_k) = \begin{bmatrix} \mathbf{k}_r & \mathbf{k}_q & \mathbf{0}_{6\times 3} & \mathbf{0}_{6\times 3} \end{bmatrix}, \tag{19}$$

$$\mathbf{A}_1 = \mathbf{k}_r\mathbf{Cov}(\mathbf{r})\mathbf{k}_r{}^T + \mathbf{k}_q\mathbf{Cov}(\mathbf{q})\mathbf{k}_q{}^T + \mathbf{a}, \tag{20}$$

where

$$\mathbf{a} = \mathbf{k}_{\mathbf{h}_d}\mathbf{Cov}(\mathbf{h}_d)\mathbf{k}_{\mathbf{h}_d}{}^T + \mathbf{k}_\rho\mathbf{Cov}(\rho)\mathbf{k}_\rho{}^T. \tag{21}$$

2.4. Measurement Modeling. The robot moves continuously and records the sequential images. This is a process of detecting and identifying the targets. The parameters of targets have been set into the process state vector and the covariance has been estimated with the recursive loop.

2.4.1. Sensor Model. The predicted pixel locations of targets are estimated as a function of the prior process state vector which is described in (4). According to (2), the location of

the ith target in the camera frame can be defined in another way:

$$\mathbf{T}_i^c = \begin{bmatrix} T_{x(i)}^c \\ T_{y(i)}^c \\ T_{z(i)}^c \end{bmatrix} = \mathbf{R}^{cw}\left(\left(\mathbf{r}_i^w - \mathbf{r}_k^w\right) * \rho_i^w + \mathbf{m}(\theta_i^w, \varphi_i^w)\right). \tag{22}$$

Our measurement sensor of the camera is monocular vision. There is a camera sensor model to describe how the sensor maps the variables in the process state vector into the sensor variables. By using the pinhole model [16], $\mathbf{T}_i^c$ is derived as a function of the undistorted pixel location of the ith target $(u_i,\ v_i)$ and denoted by

$$\mathbf{T}_i^c = \begin{bmatrix} \dfrac{(u_0 - u_i)D_x}{f} \\ \dfrac{(v_0 - v_i)D_y}{f} \\ 1 \end{bmatrix} g_{z(i)}^c \rho_i^w. \tag{23}$$

Equation (22) and (23) are combined to derive the predicted pixel location without distortion for the ith target $\mathbf{Z}_{k(i)} = (u_i, v_i)$. Its equation is addressed as

$$\mathbf{Z}_{k(i)} = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \mathbf{h}(\mathbf{Xm}_k^-) = \begin{bmatrix} u_0 - \dfrac{f}{D_x}\dfrac{T_{x(i)}^c}{T_{z(i)}^c} \\ v_0 - \dfrac{f}{D_y}\dfrac{T_{y(i)}^c}{T_{z(i)}^c} \end{bmatrix}. \tag{24}$$

By using image correct, the predicted pixel location of the ith target within distortion $\mathbf{Zd}_{k(i)} = (u_{d(i)}, v_{d(i)})$ is addressed as

$$\mathbf{Zd}_{k(i)} = \mathbf{h}_d(\mathbf{Xm}_k^-) = \begin{bmatrix} \dfrac{(u_i - u_0)}{1 + 3k_1 r_d^2 + 5k_2 r_d^4} \\ \dfrac{(v_i - v_0)}{1 + 3k_1 r_d^2 + 5k_2 r_d^4} \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix}, \tag{25}$$

where k_1 and k_2 are coefficients for the radial distortion of the image.

The actual image target the robot takes is the distorted pixel location rather than the undistorted one. Therefore, we have to calculate $\mathbf{Zd}_{k(i)}$ to get the measurement innovation for EKF updating.

2.4.2. Measurement Covariance Assignment. The measurement covariance is expected to describe what the likely variation of measurement is by the sensor under the current condition. The variation is infected by the variables in the process state vector and the noise which corrupts the sensor. The measurement Jacobian matrix $\mathbf{H}_i$ is

$$\mathbf{H}_i = \begin{bmatrix} \dfrac{\partial \mathbf{h}_{d(i)}}{\partial \mathbf{Xm}_k} & \mathbf{0}_{2\times 6} & \cdots & \dfrac{\partial \mathbf{h}_{d(i)}}{\partial \mathbf{Y}_i^w} & \mathbf{0}_{2\times 6} & \cdots & \mathbf{0}_{2\times 6} \end{bmatrix}, \tag{26}$$

where

$$\begin{aligned} \frac{\partial \mathbf{h}_{d(i)}}{\partial \mathbf{Xm}_k} &= \begin{bmatrix} \dfrac{\partial \mathbf{h}_{d(i)}}{\partial \mathbf{r}_k^w} & \dfrac{\partial \mathbf{h}_{d(i)}}{\partial \mathbf{q}_k^w} & \dfrac{\partial \mathbf{h}_{d(i)}}{\partial \mathbf{v}_k^w} & \dfrac{\partial \mathbf{h}_{d(i)}}{\partial \boldsymbol{\omega}_k^c} \end{bmatrix}_{2\times 13}, \\ \frac{\partial \mathbf{h}_{d(i)}}{\partial \mathbf{Y}_i^w} &= \begin{bmatrix} \dfrac{\partial \mathbf{h}_{d(i)}}{\partial \mathbf{r}_i^w} & \dfrac{\partial \mathbf{h}_{d(i)}}{\partial \theta_i^w} & \dfrac{\partial \mathbf{h}_{d(i)}}{\partial \varphi_i^w} & \dfrac{\partial \mathbf{h}_{d(i)}}{\partial \rho_i^w} \end{bmatrix}_{2\times 6}. \end{aligned} \tag{27}$$

When there are N targets observed concurrently, they are stacked in one measurement vector $\mathbf{Zd}_k$ to form a single batch-form update equation which is shown as

$$\mathbf{Zd}_k = \begin{bmatrix} \mathbf{Zd}_{k(1)}^T & \cdots & \mathbf{Zd}_{k(N)}^T \end{bmatrix}^T. \tag{28}$$

Similarly, the batch measurement Jacobian matrix is derived as

$$\mathbf{H_k} = \begin{bmatrix} \dfrac{\partial \mathbf{h}}{\partial \mathbf{Xm}_k} & \dfrac{\partial \mathbf{h}}{\partial \mathbf{Y}_1^w} & \cdots & \dfrac{\partial \mathbf{h}}{\partial \mathbf{Y}_N^w} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_1^T & \cdots & \mathbf{H}_N^T \end{bmatrix}^T, \tag{29}$$

where

$$\mathbf{H_k} = \begin{bmatrix} \dfrac{\partial \mathbf{h}_1}{\partial \mathbf{Xm}_k}_{2\times 13} & \dfrac{\partial \mathbf{h}_1}{\partial \mathbf{Y}_1^w}_{2\times 6} & \mathbf{0}_{2\times 6} & \mathbf{0}_{2\times 6} & \cdots & \mathbf{0}_{2\times 6} \\ \dfrac{\partial \mathbf{h}_2}{\partial \mathbf{Xm}_k}_{2\times 13} & \mathbf{0}_{2\times 6} & \dfrac{\partial \mathbf{h}_2}{\partial \mathbf{Y}_2^w}_{2\times 6} & \mathbf{0}_{2\times 6} & \cdots & \mathbf{0}_{2\times 6} \\ \dfrac{\partial \mathbf{h}_3}{\partial \mathbf{Xm}_k}_{2\times 13} & \mathbf{0}_{2\times 6} & \mathbf{0}_{2\times 6} & \dfrac{\partial \mathbf{h}_3}{\partial \mathbf{Y}_3^w}_{2\times 6} & \cdots & \mathbf{0}_{2\times 6} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial \mathbf{h}_N}{\partial \mathbf{Xm}_k}_{2\times 13} & \mathbf{0}_{2\times 6} & \mathbf{0}_{2\times 6} & \mathbf{0}_{2\times 6} & \cdots & \dfrac{\partial \mathbf{h}_N}{\partial \mathbf{Y}_N^w}_{2\times 6} \end{bmatrix}. \tag{30}$$

2.5. Image Matching by Using Cross-Correlation. The image patch information of the targets whose variables are set in the process state vector has been stored in the data-base when the robot senses them in the first time. The predicted pixel locations of the targets within distortion are estimated by using (25). Next important issue is how to search for

the actual pixel locations of the targets in a 320 × 240 image. Image matching is a fundamental and vital aspect of many problems in computer vision including motion tracking and object recognition. The image features are invariant to rotation, image scaling, change in illumination, and 3D camera viewpoint. It is still a popular research topic regarding how to allow a single candidate image to be correctly matched with high probability. Based on the estimation of measurement covariance, the small area where the target lies with high probability is able to be predicted. The measurement covariance of the ith target is defined as

$$\mathbf{Cov}\left(\mathbf{Zd}_{k(i)}\right)_{2\times 2} = \begin{bmatrix} \sigma^2_{uu(i)} & \sigma^2_{uv(i)} \\ \sigma^2_{vu(i)} & \sigma^2_{vv(i)} \end{bmatrix}. \tag{31}$$

The template match technique is used to search the actual image pixel location of the ith target in the small search area whose width and length are $k\sigma_{uu(i)}$ and $k\sigma_{vv(i)}$, respectively. The coefficient k is a constant and defines how large the search area is. The larger k is, the more is the search time. There are two advantages regarding the search area which is estimated by the measurement covariance. One advantage is that a lot of time is saved because the image pixel location of the ith target is detected in a small search area rather than in a 320 × 240 image. The other one is that the successful search rate increases dramatically because the search area allows the image pixel location to be correctly matched with high probability. As a result, it is not necessary to use a complicated object recognition algorithm such as Scale Invariant Feature Transform (SIFT) [17] for image matching in our system. Cross-correlation search is applied to be our template match algorithm and the computing loading is lower due to its simplification. This approach uses cross-correlation of image to search a suitable image patch. I_D is defined as the template image patch for the ith target and stored in the database. I_t is defined as a candidate image patch in the search area whose width is $k\sigma_{uu(i)}$ and length is $k\sigma_{vv(i)}$. The cross-correlation value of I_t with I_D is given by

$$\frac{1}{M^2-1}\sum_{uv}\frac{\left(I_D(u,v)-\overline{I}_D\right)\left(I_t(u,v)-\overline{I}_t\right)}{\sigma_{I_D}\sigma_{I_t}}. \tag{32}$$

M^2 is the number of pixels in an image patch and σ_{I_D} and σ_{I_t} are the stand deviations of I_D and I_D, respectively. $\overline{I}_D$ and $\overline{I}_t$ are the average values of I_D and I_t, respectively. The maximum value of cross-correlation of I_t with I_D is the best template matching and it is seen as the matching target pixel patch for the ith target.

2.6. Iterated EKF Updating. The EKF is one of the most widely used nonlinear estimators due to its similarity to the optimal linear filter, its simplicity of implementation, and its ability to provide accurate estimates in practice. We employ the iterated EKF to update the state. At each iteration step k, the prior process state vector is computed by using (4) and then $\hat{\mathbf{Z}}\mathbf{d}_k$ is calculated as a function of $\mathbf{Xm}_k^-$ by using (28). Next, the measurement innovation and the measurement Jocobian matrix are computed as $\mathbf{Zd}_k - \hat{\mathbf{Z}}\mathbf{d}_k$ and $\mathbf{H}_{\mathbf{k}(2N\times(13+6N))}$, respectively. The measurement covariance and Kalman gain can be performed as

$$\begin{aligned} \mathbf{Cov}(\mathbf{Zd}_k)_{2N\times 2N} &= \mathbf{H}_k\mathbf{Cov}(\mathbf{X}_k^-)_{(13+6N)\times(13+6N)}\mathbf{H}_k^T, \\ \mathbf{K}_{k(13+6N)\times 2N} &= \mathbf{Cov}(\mathbf{X}_k^-)\mathbf{H}_k^T[\mathbf{Cov}(\mathbf{Zd}_k)]^{-1}. \end{aligned} \tag{33}$$

Finally, the process state vector and its covariance are updated at each iteration state and presented as

$$\begin{aligned} \mathbf{X}_{k((13+6N)\times 1)} &= \mathbf{X}_k^- + \mathbf{K}_k\left[\mathbf{Zd}_{k(2N\times 1)} - \hat{\mathbf{Z}}\mathbf{d}_{k(2N\times 1)}\right], \\ \mathbf{Cov}(\mathbf{X}_k)_{(13+6N)\times(13+6N)} &= \mathbf{Cov}\left(\mathbf{X}_k^-\right) - \mathbf{K}_k\mathbf{H}_k\mathbf{Cov}\left(\mathbf{X}_k^-\right), \end{aligned} \tag{34}$$

where $\mathbf{Zd}_k$ is the actual measurement and detected by using image matching search. $\hat{\mathbf{Z}}\mathbf{d}_k$ is the predicted measurement and computed by the sensor model. The error in the estimation is reduced by the iteration and the unknown depths of the targets converge toward the real values gradually.

2.7. Fast Inverse Transformation for Covariance Matrices. The inverse matrix of the measurement covariance has to be computed at each iteration step. It needs plenty of running time by using the transpose of the matrix of cofactors to invert the measurement covariance. In order to deal with the large size of the inverse matrix within the variations of N targets efficiently, Cholesky decomposition is applied to invert the measurement covariance and reduce the running time at each iteration step. The measurement covariance is factored in the form $\mathbf{Cov}(\mathbf{Zd}_k) = \mathbf{LL}^T$ for a lower-left corner of the matrix $\mathbf{L}$ because the measurement covariance is positive semidefinite. The matrix $\mathbf{L}$ is not unique, and so multiple factorizations of a given matrix $\mathbf{Cov}(\mathbf{Zd}_k)$ are possible. A standard method for factoring positive semidefinite matrices is the Cholesky factorization. The element of $\mathbf{L}$ is computed by using Cholesky decomposition and addressed as follows:

$$\begin{aligned} L_{jj} &= \sqrt{A_{jj} - \sum_{k=1}^{j-1} L_{jk}^2}, \\ L_{ij} &= \frac{1}{L_{jj}}\left(A_{ij} - \sum_{k=1}^{j-1} L_{ik}L_{jk}\right), \end{aligned} \tag{35}$$

where A_{ij} is the element of $\mathbf{A} = \mathbf{C}(\mathbf{Zd}_k)$.

A matrix equation in the form $\mathbf{LX} = \mathbf{b}$ or $\mathbf{UX} = \mathbf{b}$ can be solved by forward substitution for lower triangular matrix $\mathbf{L}$ and back substitution for upper triangular matrix $\mathbf{U}$, respectively. In our proposed method, Cholesky decomposition and forward-and-back substitution are combined to invert the measurement covariance for reducing the computational loading.

3. The Analysis of the Algorithm

In this section, there are two key points about the proposed algorithm presented as follows. The first key point is that

the details of algorithm will be analyzed to prove that it is not necessary to get the depth information in our algorithm even though it is designed to track the 3D localization of the targets. The other key point is introduced to explain why the simple template match technique, cross-correlation search, is applied.

3.1. Depth Information Analysis in the Proposed Algorithm. The depth information should be known while the robot localizes the targets in 3D space by using monocular vision. However, the proposed algorithm does not need the depth information to localize targets. As presented in (14), it is not necessary to know the depth information to calculate θ_1^w and φ_1^w. Similarly, it seems that the depth information should be known to compute $\mathbf{k}_q$ in (19). $\mathbf{k}_q$ is derived as

$$\mathbf{k}_q = \left[\frac{\partial \mathbf{r}_i^{w\,T}}{\partial \mathbf{q}_k^w} \quad \frac{\partial \theta_i^{w\,T}}{\partial \mathbf{q}_k^w} \quad \frac{\partial \varphi_i^{w\,T}}{\partial \mathbf{q}_k^w} \quad \frac{\partial \rho_i^{w\,T}}{\partial \mathbf{q}_k^w} \right]_{6\times 4}^{T}, \tag{36}$$

where

$$\frac{\partial \theta_i^w}{\partial \mathbf{q}_k^w} = \frac{\partial \theta_i^w}{\partial \mathbf{g}^w} * \frac{\partial \mathbf{g}^w}{\partial \mathbf{q}_k^w}. \tag{37}$$

The depth information g_z^w has to be known for calculating $\partial\theta_i^w/\partial \mathbf{g}^w$ according to the following equation:

$$\frac{\partial \theta_i^w}{\partial \mathbf{g}^w} = \left[\frac{g_z^w}{(g_x^w)^2 + (g_z^w)^2} \quad 0 \quad \frac{-g_x^w}{(g_x^w)^2 + (g_z^w)^2} \right]. \tag{38}$$

However, $\partial\theta_i^w/\partial \mathbf{q}_k^w$ still can be computed without knowing g_z^w because of $\partial \mathbf{g}^w/\partial \mathbf{q}_k^w$. The details are proved as follows:

$$\frac{\partial \theta_i^w}{\partial \mathbf{q}_k^w} = \left[\frac{g_z^w/g_z^c}{(g_x^w/g_z^c)^2 + (g_z^w/g_z^c)^2} \quad 0 \quad \frac{-g_x^w/g_z^c}{(g_x^w/g_z^c)^2 + (g_z^w/g_z^c)^2} \right] \times \frac{1}{g_z^c} \frac{\partial \mathbf{g}^w}{\partial \mathbf{q}_k^w}, \tag{39}$$

where

$$\frac{1}{g_z^c} \frac{\partial \mathbf{g}^w}{\partial \mathbf{q}_k^w} = \left[\frac{\partial \mathbf{R}^{wc}}{\partial q_1} \frac{\mathbf{g}^c}{g_z^c} \quad \frac{\partial \mathbf{R}^{wc}}{\partial q_2} \frac{\mathbf{g}^c}{g_z^c} \quad \frac{\partial \mathbf{R}^{wc}}{\partial q_3} \frac{\mathbf{g}^c}{g_z^c} \quad \frac{\partial \mathbf{R}^{wc}}{\partial q_4} \frac{\mathbf{g}^c}{g_z^c} \right]. \tag{40}$$

g_x^w/g_z^c, g_y^w/g_z^c, and g_z^w/g_z^c are computed by (14). Equation (39) can be calculated without knowing the depth information g_z^c due to the following equation:

$$\frac{\mathbf{g}^c}{g_z^c} = \begin{bmatrix} \dfrac{(u_0 - u)D_x}{f} \\ \dfrac{(v_0 - v)D_y}{f} \\ 1 \end{bmatrix}. \tag{41}$$

In the same way, the depth information should be known to compute $\mathbf{k}_{\mathbf{h}_d}$ in equation (19). $\mathbf{k}_{\mathbf{h}_d}$ is derived as

$$\mathbf{k}_{\mathbf{h}_d} = \left[\frac{\partial \mathbf{r}_i^{w\,T}}{\partial \mathbf{h}_d} \quad \frac{\partial \theta_i^{w\,T}}{\partial \mathbf{h}_d} \quad \frac{\partial \varphi_i^{w\,T}}{\partial \mathbf{h}_d} \quad \frac{\partial \rho_i^{w\,T}}{\partial \mathbf{h}_d} \right]_{6\times 2}^{T}, \tag{42}$$

Figure 3: Detect the actual target pixel location by using cross-correlation search.

where

$$\frac{\partial \theta_i^w}{\partial \mathbf{h}_d} = \frac{\partial \theta_i^w}{\partial \mathbf{g}^w} \frac{\partial \mathbf{g}^w}{\partial \mathbf{g}^c} \frac{\partial \mathbf{g}^c}{\partial \mathbf{h}_u} \frac{\partial \mathbf{h}_u}{\partial \mathbf{h}_d}. \tag{43}$$

$\partial \mathbf{g}^w/\partial \mathbf{g}^c = \mathbf{R}^{wc}$ and $\partial \mathbf{h}_u/\partial \mathbf{h}_d$ can be computed without g_z^c but $\partial\theta_i^w/\partial \mathbf{g}^w$ cannot be calculated without g_z^w according to (39). However, $\partial\theta_i^w/\partial \mathbf{h}_d$ still can be estimated without knowing g_z^c and g_z^w if $\partial\theta_i^w/\partial \mathbf{g}^w$ and $\partial \mathbf{g}^c/\partial \mathbf{h}_u$ are combined. The details are proved by the following equations:

$$\frac{\partial \theta_i^w}{\partial \mathbf{g}^w} \frac{\partial \mathbf{g}^c}{\partial \mathbf{h}_u} = \left[\frac{g_z^w/g_z^c}{(g_x^w/g_z^c)^2 + (g_z^w/g_z^c)^2} \quad 0 \quad \frac{-g_x^w/g_z^c}{(g_x^w/g_z^c)^2 + (g_z^w/g_z^c)^2} \right] \times \frac{1}{g_z^c} * \frac{\partial \mathbf{g}^c}{\partial \mathbf{h}_u}, \tag{44}$$

where

$$\frac{\partial \mathbf{g}^c}{\partial \mathbf{h}_u} = \begin{bmatrix} \dfrac{-D_x}{f} & 0 \\ 0 & \dfrac{-D_y}{f} \\ 0 & 0 \end{bmatrix} g_z^c. \tag{45}$$

Therefore, $\partial\theta_i^w/\partial \mathbf{h}_d$ is computed without g_z^c and g_z^w by using (44) and applying (14) to calculate g_x^w/g_z^c, g_y^w/g_z^c and g_z^w/g_z^c.

Based on the results of calculating $\partial\theta_i^w/\partial \mathbf{q}_k^w$, $\partial\theta_i^w/\partial \mathbf{h}_d$ and (14), it infers that we have solved a problem that the depth information, g_z^c and g_z^w, is not able to be measured only by one image. This is a very important characteristic of the proposed algorithm.

3.2. Analysis Regarding Using Cross-Correlation Search. Generally speaking, cross-correlation search is though as a simple template match algorithm and it is not as robust as SIFT. However, I_D is still detected correctly in the search area by using cross-correlation search because the small area is estimated by an iterated EKF and it includes the actual pixel location of the target with high probability. As shown in Figure 3, the predicted pixel locations of the targets are estimated by using the sensor model and denoted by the green crosses. The pixel locations of the red crosses are the corresponding target pixel locations and detected by applying cross-correlation search. Based on the testing result that the red crosses are closer to the actual target pixel locations, it has proved that it is feasible for the proposed algorithm to apply cross-correlation search as its template match algorithm.

(a) (b) (c)

Figure 4: The encoder shown in (a) is used to measure the linear velocity and simulate the acceleration for the robot. The electrical compass shown in (b) is applied to measure the angular velocity of the robot. As shown in (c), the camera with wide angle lens is the additional measurement sensor. The three sensors are combined to estimate the localization of targets.

4. Experimental Results

In order to validate the proposed algorithm for localizing targets when the ground truth is available we have performed a number of experiments. The algorithm is implemented by C++ and performed by PC with 2.0 GHz microprocessor. The monocular sensor is a single camera with wide angle lens because we hope that more targets can be observed in one image and tracking rate can be higher. The camera's field of view is 170° with a focal length of 1.7 mm. The image measurements received at a rate of 15 Hz are distorted with noise $\sigma = 20$ pixel. The addressed experimental results are tested under the high speed robot motion because the average velocity of each case is higher than 20 cm/sec and the maximum velocity of all cases is 69.11 cm/sec. For the duration of experiments, the initial distance between the camera and the targets ranges from 1.68 m to 5.76 m. The unknown depth of the target is estimated by sequential images with EKF and six cases (3.0 m, 3.5 m, 4.0 m, 4.5 m, 5.0 m, and 5.5 m) are assumed to be default depths for each experiment case. All of the sensors mounted on our system are shown in Figure 4.

There are some measurement errors caused by the camera distortion when using the camera with wide angle lens. Before validating the proposed algorithm, we performed an experiment to estimate the distorted noise by making use of the artificial landmarks. We chose the corners of the artificial landmarks as targets. The undistorted pixel locations of the corners are estimated by the pinhole model and then the image correction is applied to compute their predicted pixel locations with distortion. Owing to using a camera with wide angle lens, there is an error between the predicted pixel location with distortion and its actual pixel location which is detected by the cross-correlation search. In the proposed algorithm, the predicted pixel location without distortion is estimated in terms of the prior process state vector. The distorted error is produced if transforming the undistorted pixel locations of targets to the distorted pixel locations. Therefore, the distorted error should be taken into consideration very carefully. Based on the experimental result, the distorted noise is assumed to be 20 pixels.

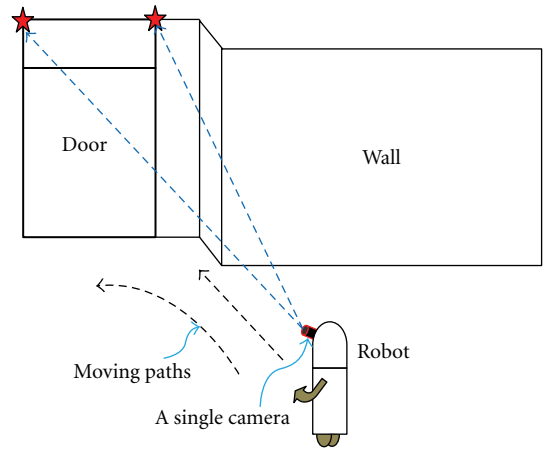

Figure 5: The experiments are designed to localize the natural landmarks as targets with varied linear and angular velocity patterns in different moving paths including the straight and curved paths.

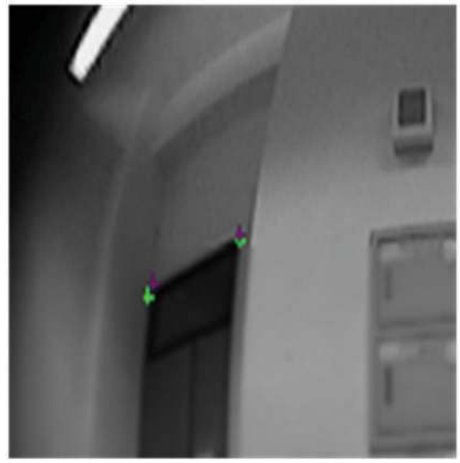

Figure 6: The predicted pixel locations of the targets are denoted by the green crosses and estimated by using the sensor model. The pixel locations of the red crosses are the corresponding target pixel locations and detected by applying cross-correlation search.

4.1. Results of Target Localization. For demonstrating the performance and practical applications to the proposed MVLT, the experiments for a mobile robot go through the doors by tracking the feature targets located upon the door and localize robot location simultaneously.

4.1.1. Target Localization Experiments. Once a robot has to pass through doors, one should recognize the accurate doors position. The experiment shown in Figure 5 is designed to verify accuracy of MVLT for the door as tracked targets. Figure 6 shows the real environmental features as targets whose positions require to be estimated for a task that the robot passing through the door. Since the developed system does not require an initial setting at the first iteration, it provides a practical usage under a dynamical motion if target is detected instead of training processes of the environment features for robot localization. The two different moving

Table 1: Error comparison of target tracking.

Algorithm	Item	Estimated target position (m)			Track OK	Target distance error (m)
		X	Y	Z		
MVTL	Target 1 in Path A	0.7600	0.1423	0.6528	OK	0.1369
	Target 2 in Path A	0.7800	−0.0374	0.6909	OK	0.1303
	Target 1 in Path B	0.8100	0.0975	0.6853	OK	0.0839
	Target 2 in Path B	0.7600	−0.1135	0.5819	OK	0.1523
MVTL Average Error						0.1049
MonoSLAM [18]	Target 1 in Path A	0.7800	0.0460	0.8234	OK	0.2042
	Target 2 in Path A	0.8370	−0.0218	1.0201	OK	0.3723
	Target 1 in Path B	1.1158	0.0896	1.2330	Failed	0.6160
	Target 2 in Path B	1.2281	−0.1161	1.4326	Failed	0.8435
MonoSLAM [18] Average Error						0.5090

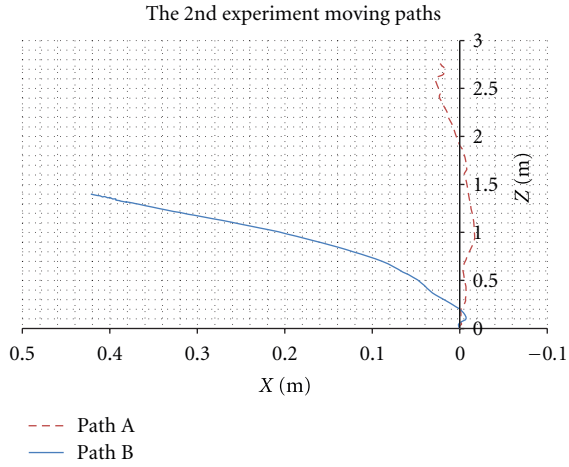

Figure 7: Ground truth of experiments.

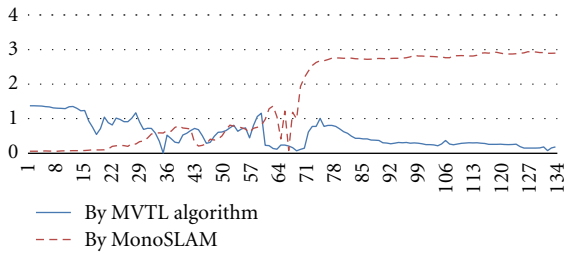

Figure 8: The comparison of the target 1 depth errors (meter) between MVTL and MonoSLAM [18] in Path B.

paths with varied linear velocities and orientations are examined as shown in Figure 7. The initial true depth for paths A and B between robot and the doors is about 3.5 meter, path A is a straight forward path, and path B is a left way of robot to move forward to the doors. The experimental result is summarized in Table 1, and the averaged error of proposed approach is about 10 cm.

According to the experimental result shown in Figures 8 and 9, it infers that the camera is not able to provide the

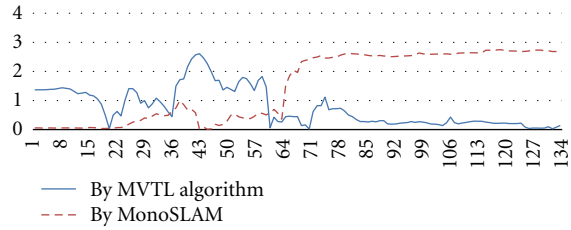

Figure 9: The comparison of the target 2 depth errors (meter) between MVTL and MonoSLAM [18] in Path B.

precise measurement information to make MonoSLAM [18] correcting the prior process state variables under a high-speed movement when the control input is not provided. It also shows that the image-distorted noise $\sigma = 20$ pixel is too large for MonoSLAM [18] model. Owing to combining hybrid sensor information, the proposed MVTL algorithm is capable of localizing targets with image distorted noise $\sigma = 20$ pixel and its average error is lower than 0.11 m. It has proved that MVTL algorithm is robust to track targets under a higher-speed movement with larger measurement noise.

4.1.2. Acceleration Approximation-Based Error Comparison. We also performed an experiment about the comparison between encoders and IMU. We choose an encoder as our sensor instead of IMU in indoor environment. The acceleration $\mathbf{a}^{w}_{(k)}$ is approximated as $(\mathbf{v}^{\mathbf{w}}_{(\mathbf{k})} - \mathbf{v}^{\mathbf{w}}_{(\mathbf{k-1})})/\Delta t$ by encoders in order to have similar on-line simulation data. It will increase additional acceleration noise if we use prevelocity $\mathbf{v}^{\mathbf{w}}_{(\mathbf{k})}$ rather than $\mathbf{v}^{\mathbf{w}}_{(\mathbf{k+1})}$ to simulate the acceleration of the robot at state k. However, an iterated EKF is robust enough to compensate for errors from this low-cost sensor and the definition of acceleration. The error of acceleration from a simulated IMU is lower than 0.001 m/s^2. Based on the result shown in Table 2, the errors of localization by using encoders are still accepted even though we use prevelocity $\mathbf{v}^{\mathbf{w}}_{(\mathbf{k})}$ to simulate acceleration at iterated state k. It has proved that errors are able to be reduced gradually with EKF recursive loop.

Table 2: Comparison of localization errors between the encoder and IMU.

Robot motion	Maximum Velocity (m/sec)	Average Velocity (m/sec)	IMU Error (m)	Encoder Error (m)
Low Velocity	0.363	0.254	0.052	0.089
High Velocity	0.582	0.446	0.142	0.144

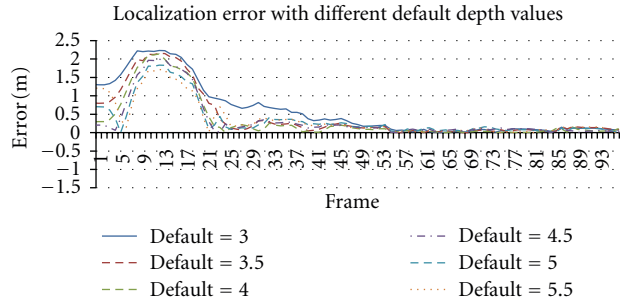

Figure 10: The unknown depth of the target is estimated by sequential images with EKF and six cases (3.0 m, 3.5 m, 4.0 m, 4.5 m, 5.0 m, and 5.5 m) are assumed to be default depths for each experiment case.

4.1.3. Performance of Depth Initial Conditions. In terms of the experimental result shown in Figure 10, we could conclude that the ability of localizing targets depends on the parallax from different views rather than the distance between the robot and the target. It is a very crucial viewpoint to analyze the stability of the system in terms of means and deviations. Not only do we analyze one the target localization but also we understand the details about multiple target localization. Refering to the above experiments, MVTL algorithm localizes targets with a higher-speed motion and six kinds of default depth values converge to the similar value with little errors.

4.2. Experiments of Fast Inverse Matrix Transformation. Finally, we performed an experiment to verify Cholesky decomposition and forward-and-back substitution. In this experiment, the totally testing time is 4.2 sec with 15 Hz image rate. At first, the transpose of the matrix of cofactors is used to invert the measurement covariance matrix whose size ranges from 2×2 to 8×8. The experimental results shown in Table 3 present that the running time is long if the system inverts a 6×6 or a more dimension matrix. However, the running time is reduced dramatically if the system inverts an 8×8 matrix by using Cholesky decomposition and forward-and-back substitution. The test result shows that the system is capable of being real-time even though localizing four targets concurrently.

5. Conclusion and Future Work

5.1. Conclusion. Based on the different experiments we performed, it has proved that the proposed algorithm is able to localize targets with "cm" level accuracy under a higher-speed movement. Besides, we have validated that it is practical to use odometry sensors to track targets as well. Not only does the system start the recursive procedure without the initial setting but also the robot can move rapidly to localize targets. The EKF-based algorithm is really practical and robust to reduce errors from sensors even though the low-cost sensors are used in our system. The efficiency of the proposed algorithm is impressive by using Cholesky decomposition and some computational tricks. In terms of the experimental result shown in Figure 10, we could conclude that the ability of localizing targets depends on the parallax from different views rather than the distance between the robot and the target. Consequently, the proposed algorithm has a high potential extension to surveillance and monitoring for UAVs with aerial odometry sensors. In the same way, it is able to be used widely for robot tasks as well.

Table 3: Comparison of execution time.

Target number	1	2	3	4
Cholesky decomposition (sec)	0.877	1.182	1.548	1.92
Matrix of cofactors (sec)	0.871	1.216	7.07	516.742

5.2. Future Work. The targets are assumed to be stationary landmarks in the proposed algorithm. It will be an interesting and challenging research if the algorithm is modified to track a moving target. This type of the technique is necessary and going to be widely used in many applications. For instance, UGV has to know the intended motions of other moving objects in order to plan its navigating path for the next state. Therefore, it will be a good future work to add a new approach into the proposed algorithm to track moving targets. Besides, there is another important future work for our system. This future work is to find some ways or algorithms to improve accuracy of the measurement data by the low-cost sensor because it will improve the localization errors in terms of more accurate measurement data.

Appendices

The proof of the system covariance is presented in appendix. It derives the modified covariance in motion model at each iteration step and the new covariance after adding new targets.

A. The Modified System Covariance in Motion Model

The process state vector is predicted by the system process model. It is determined by the old state and the control inputs applied in the process. $\mathbf{Y}_k$ is a control vector and

u is corrupted by mean-zero process. Both of equations are described as

$$\begin{aligned} \mathbf{Y} &= \mathbf{u} + \mathbf{w}, \\ \mathbf{Cov}(\mathbf{Y}) &= \mathbf{Cov}(\mathbf{w}). \end{aligned} \tag{A.1}$$

The prior estimate error covariance $\mathbf{Cov}(\mathbf{X}_k^-)$ is derived as a function of the prior state estimate $\hat{\mathbf{X}}_k^-$ and represented as

$$\mathbf{Cov}\left(\mathbf{X}_k^-\right) \approx \mathbf{F}_{(x,y)}\mathbf{C}\mathbf{F}_{(x,y)}{}^T, \tag{A.2}$$

where

$$\begin{aligned} \mathbf{C} &= \begin{bmatrix} \mathbf{Cov}(\mathbf{X}_{k-1}) & \mathbf{Cov}(\mathbf{X}_{k-1}, \mathbf{Y}_{k-1}) \\ \mathbf{Cov}(\mathbf{Y}_{k-1}, \mathbf{X}_{k-1}) & \mathbf{Cov}(\mathbf{Y}_{k-1}) \end{bmatrix}, \\ \mathbf{F}_{(X,Y)} &= \begin{bmatrix} \mathbf{F}_X & \mathbf{F}_Y \end{bmatrix} \triangleq \frac{\partial \mathbf{f}(\mathbf{X}, \mathbf{Y})}{\partial(\mathbf{X}, \mathbf{Y})}\left(\hat{\mathbf{X}}_{k-1}, \hat{\mathbf{Y}}_{k-1}\right). \end{aligned} \tag{A.3}$$

If the control vector is not correlated with the state $\mathbf{X}_{k-1}$, $\mathbf{Cov}(\mathbf{X}_{k-1}, \mathbf{Y}_{k-1})$ and $\mathbf{Cov}(\mathbf{Y}_{k-1}, \mathbf{X}_{k-1})$ are equal to zero matrices. The prior estimate error covariance $\mathbf{Cov}(\mathbf{X}_k^-)$ simplifies to

$$\mathbf{Cov}\left(\mathbf{X}_k^-\right) \approx \mathbf{F}_X\mathbf{Cov}(\mathbf{X}_{k-1})\mathbf{F}_X^T + \mathbf{F}_Y\mathbf{Cov}(\mathbf{Y}_{k-1})\mathbf{F}_Y^T. \tag{A.4}$$

B. The New Covariance after Adding New Targets

When the robot tracks a target, the location information of new target is added into the process state vector. The form of the new information is determined by sensor measurement model. $\hat{\mathbf{x}}_{y_1}$ is composed of the random variables of the new target information. The new covariance is addressed as

$$\mathbf{Cov}(\mathbf{X}_{\text{new}}) = \begin{bmatrix} \mathbf{Cov}(\mathbf{X}_{\text{old}}) & \mathbf{B}^T \\ \mathbf{B} & \mathbf{A} \end{bmatrix}. \tag{B.1}$$

The size of the posterior estimate error covariance $\mathbf{Cov}(\mathbf{X}_k)$ increases with the number of the targets (Figure 11). The EFK is a multihypothesis and a proper way to include all the measurement information. There are two cases regarding the relationship between the new target and other variables. One case is that the estimate of the new target's location is independent of the estimates of other targets and the variables of the robot. In this case, the covariance matrix of the new target is given by

$$\begin{aligned} \mathbf{A} &= \mathbf{Cov}\left(\mathbf{x}_{\mathbf{y}_1}\right), \\ \mathbf{B}_i &= \mathbf{Cov}\left(\mathbf{x}_{\mathbf{y}_1}, x_i\right) = \mathbf{0}. \end{aligned} \tag{B.2}$$

A is a covariance matrix and **B** is a cross-covariance matrix. **B** is a zero matrix because the new target is independent of other targets and the robot by definition. The other case is that the new target is determined as a function **g** of its spatial relation **z** to other target locations. The new target's location is correlated with the estimates of other targets and the robot.

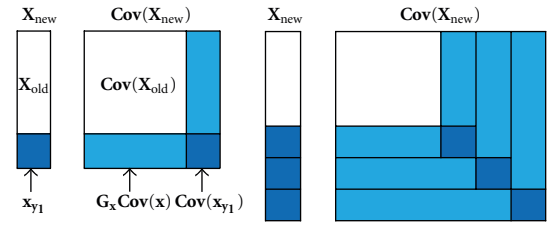

FIGURE 11: The size of the posterior estimate error covariance $\mathbf{Cov}(\mathbf{X}_k)$ increases with the number of the targets. The size of the system covariance is modified from a 13×13 matrix to a 19×19 matrix while the robot senses one target in the process state vector.

The function **g** and the covariance matrix of the new target are shown as

$$\mathbf{x}_{\mathbf{y}_1} = \mathbf{g}(\mathbf{x}, \mathbf{z}), \tag{B.3}$$

$$\mathbf{A} = \mathbf{G}_x\mathbf{Cov}(\mathbf{x})\mathbf{G}_x^T + \mathbf{G}_z\mathbf{Cov}(\mathbf{z})\mathbf{G}_z^T, \tag{B.4}$$

$$\mathbf{B}_i = \mathbf{G}_x\mathbf{Cov}(\mathbf{x}). \tag{B.5}$$

Acknowledgment

This work is supported by the National Science Council under Grant no. NSC 99-2221-E-009-159.

References

[1] G. Cicirelli, T. D'Orazio, and A. Distante, "Target recognition by components for mobile robot navigation," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 15, no. 3, pp. 281–297, 2003.

[2] S. Sohn, B. Lee, J. Kim, and C. Kee, "Vision-based real-time target localization for single-antenna GPS-guided UAV," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 4, pp. 1391–1401, 2008.

[3] P. Pérez, J. Vermaak, and A. Blake, "Data fusion for visual tracking with particles," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 495–513, 2004.

[4] F. M. Mirzaei and S. I. Roumeliotis, "A Kalman filter-based algorithm for IMU-camera calibration: observability analysis and performance evaluation," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1143–1156, 2008.

[5] S. B. Kim, S. Y. Lee, J. H. Choi, K. H. Choi, and B. T. Jang, "A bimodal approach for GPS and IMU integration for land vehicle applications," in *Proceedings of the 58th IEEE Vehicular Technology Conference (VTC '03)*, pp. 2750–2753, October 2003.

[6] K. Park, H. Chung, J. Choi, and J. G. Lee, "Dead reckoning navigation for an autonomous mobile robot using a differential encoder and a gyroscope," in *Proceedings of the 8th International Conference on Advanced Robotics (ICAR '97)*, pp. 441–446, July 1997.

[7] S. Suksakulchai, S. Thongchai, D. M. Wilkes, and K. Kawamura, "Mobile robot localization using an electronic compass for corridor environment," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 3354–3359, October 2000.

[8] J. Schroeder, S. Galler, K. Kyamakya, and K. Jobmann, "Practical considerations of optimal three-dimensional indoor localization," in *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI '06)*, pp. 439–443, September 2006.

[9] S. Emura and S. Tachi, "Sensor fusion based measurement of human head motion," in *Proceedings of the 3rd IEEE International Workshop on Robot and Human Communication*, pp. 124–129, July 1994.

[10] P. Grandjean and A. Robert De Saint Vincent, "3-D modeling of indoor scenes by fusion of noisy range and stereo data," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, pp. 681–687, May 1989.

[11] P. S. Maybeck, *Stochastic Models Estimation and Control*, vol. 1, Academic Press, 1979.

[12] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, The MIT Press, 2005.

[13] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*, The MIT Press, 2004.

[14] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 4, p. 850, 1987.

[15] D. Dereniowski and M. Kubale, "Cholesky factorization of matrices in parallel and ranking of graphs," in *Proceedings of the 5th International Conference on Parallel Processing and Applied Mathematics*, vol. 3019 of *Lecture Notes on Computer Science*, pp. 985–992, Springer, 2004.

[16] J. Heikkila and O. Silven, "Four-step camera calibration procedure with implicit image correction," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1106–1112, June 1997.

[17] S. Se, D. Lowe, and J. Little, "Vision-based mobile robot localization and mapping using scale-invariant features," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '01)*, vol. 2, pp. 2051–2058, May 2001.

[18] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.