EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

Detection of moving objects in image plane for robot navigation using monocular vision

Yin-Tien Wang*, Chung-Hsun Sun and Ming-Jang Chiou

Abstract

This article presents an algorithm for moving object detection (MOD) in robot visual simultaneous localization and mapping (SLAM). This MOD algorithm is designed based on the defining epipolar constraint for the corresponding feature points on image plane. An essential matrix obtained using the state estimator is utilized to represent the epipolar constraint. Meanwhile, the method of speeded-up robust feature (SURF) is employed in the algorithm to provide a robust detection for image features as well as a better description of landmarks and of moving objects in visual SLAM system. Experiments are carried out on a hand-held monocular camera to verify the performances of the proposed algorithm. The results show that the integration of MOD and SURF is efficient for robot navigating in dynamic environments.

Keywords: simultaneous localization, and mapping (SLAM), moving object detection (MOD), moving object tracking (MOT), speeded-up robust features (SURF), monocular vision

1. Introduction

In recent years, more and more researchers solve the simultaneous localization and mapping (SLAM) as well as the moving object tracking (MOT) problems concurrently. Wang et al. [1] developed a consistency-based moving object detector and provided a framework to solve the SLAMMOT problems. Bibby and Reid [2] proposed a method that combines sliding window optimization and least-squares together with expectation maximization to do reversible model selection and data association that allows dynamic objects to be included directly into the SLAM estimation. Zhao et al. [3] used GPS data and control inputs to achieve global consistency in dynamic environments. There are many advantages to cope with SLAM and MOT problems simultaneously: for example, mobile robots might navigate in a dynamic environment crowded with moving objects. In this case the SLAM could be corrupted with the inclusion of moving entities if the information of moving objects is not taken account. Furthermore, the robustness of robot localization and mapping algorithms can be improved if the moving objects are discriminated from the stationary objects in the environment.

Using cameras to implement SLAM is the current trend because of their light weight and low-cost features, as well as containing rich appearance and texture information of the surroundings. However, it is still a difficult problem in visual SLAM to discriminate the moving objects from the stationary landmarks in dynamic environments. To deal with this problem, we propose the moving object detection (MOD) algorithm based on the epipolar constraint for the corresponding feature points on image plane. Given an estimated essential matrix it is possible to investigate whether a set of corresponding image points satisfy the defining epipolar constraint in image plane. Therefore, the epipolar constraint can be utilized to distinguish the moving objects from the stationary landmarks in dynamic environments.

For visual SLAM systems, the features in the environment are detected and extracted by analyzing the image taken by the robot vision, and then the data association between the extracted features and the landmarks in the map is investigated. Many researchers [4,5] employed the concept by Harris and Stephens [6] to extract apparent corner features from one image and tracked these point features in the consecutive image. The descriptors of the Harris corner features are rectangle image patches. When the camera translates and rotates, the

* Correspondence: ytwang@mail.tku.edu.tw
Department of Mechanical and Electro-Mechanical Engineering, Tamkang University, Tamsui, New Taipei City 25137, Taiwan

scale and orientation of the image patches will be changed. The detection and matching of Harris corner might fail in this case, unless the variances in scale and orientation of the image patches are recovered. Instead of detecting corner features, some works [7,8] detect the features by using the scale-invariant feature transform (SIFT) method [9] which provides a robust image feature detector. The unique properties of image features extracted by SIFT method are further described by using a high-dimensional description vector [9]. However, the feature extraction by SIFT requires more computational cost than that by Harris's method [6]. To improve the computational speed, Bay et al. [10] introduced the concept of integral images and box filter to detect and extract the scale-invariant features, which they dubbed speeded-up robust features (SURF). The extracted SURF must be matched with the landmarks in the map of a SLAM system. The nearest-neighbor (NN) searching method [11] can be utilized to match high-dimensional data sets of description vectors.

In this article, an online SLAM system with a moving object detector is developed based on the epipolar constraint for the corresponding feature points on image plane. The corresponding image features are obtained using the SURF method [10] and the epipolar constraint is calculated using an estimated essential matrix. Moving object information is detected in image plane and integrated into the MOT process such that the robustness of SLAM algorithm can be considerably improved, particularly in highly dynamic environments where surroundings of robots are dominated by non-stationary objects. The contributions in this article are twofold. First, we develop an algorithm to solve the problems for MOD in image plane, and then the algorithm is integrated with the robot SLAM to improve the robustness of state estimation and mapping processes. Second, the improved SLAM system is implemented on a hand-held monocular camera which can be utilized as the sensor system for robot navigation in dynamic environments.

The SLAM problem with monocular vision will be briefly introduced in Section 2. In Section 3, the proposed algorithm of MOD is explained in detail. Some examples to verify the performance of the data association algorithm are described in Section 4. Section 5 is the concluding remarks.

2. SLAM with a free-moving monocular vision

SLAM is a target tracking problem for the robot system during navigating in the environment [12]. The targets to be tracked include the state of the robot itself as well as of the landmarks and moving objects in the environment. The state sequence of the SLAM system at time step k can be expressed as

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, w_{k-1}) \tag{1}$$

where $\mathbf{x}_k$ is the state vector; $\mathbf{u}_k$ is the input; w_k is the process noise. The objective of the tracking problem is to recursively estimate the state $\mathbf{x}_k$ of the target according to the measurement $\mathbf{z}_k$ at time step k,

$$\mathbf{z}_k = g(\mathbf{x}_k, v_k) \tag{2}$$

where v_k is the measurement noise. A hand-held monocular vision, as shown in Figure 1, is utilized in this article as the only sensing device for the measurement in SLAM system. We treat this hand-held vision sensor as a free-moving robot system with unknown inputs. The states of the system are estimated by solving the recursive SLAM problem using the extended Kalman filter (EKF) [12]

$$\mathbf{x}_{k|k-1} = f(\mathbf{x}_{k-1|k-1}, \mathbf{u}_{k-1}, 0) \tag{3a}$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{A}_k \boldsymbol{P}_{k-1|k-1} \boldsymbol{A}_k^T + \boldsymbol{W}_k \boldsymbol{Q}_{k-1} \boldsymbol{W}_k^T \tag{3b}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^T (\boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^T + \boldsymbol{V}_k \boldsymbol{R}_k \boldsymbol{V}_k^T)^{-1} \tag{3c}$$

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \boldsymbol{K}_k (\mathbf{z}_k - g(\mathbf{x}_{k|k-1}, 0)) \tag{3d}$$

$$\boldsymbol{P}_{k|k} = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \boldsymbol{P}_{k|k-1} \tag{3e}$$

where $\mathbf{x}_{k|k-1}$ and $\mathbf{x}_{k|k}$ represent the predicted and estimated state vectors, respectively; $\boldsymbol{K}_k$ is Kalman gain matrix; $\boldsymbol{P}$ denotes the covariance matrix, respectively; $\boldsymbol{A}_k$ and $\boldsymbol{W}_k$ are the Jacobian matrices of the state equation f with respect to the state vector $\mathbf{x}_k$ and the noise variable w_k, respectively; $\boldsymbol{H}_k$ and $\boldsymbol{V}_k$ are the Jacobian matrices of the measurement g with respect to the state vector $\mathbf{x}_k$ and the noise variable v_k, respectively.

2.1. Motion model

Two coordinate systems are set at the world frame $\{W\}$ and the camera frame $\{C\}$, as shown in Figure 2. The state vector of the SLAM system with MOT in Equation (1) is arranged as

$$\mathbf{x} = [\mathbf{x}_C \ \mathbf{m}_1 \ \mathbf{m}_2 \ \cdots \ \mathbf{m}_n \ \mathbf{O}_1 \ \mathbf{O}_2 \ \cdots \ \mathbf{O}_l]^T \tag{4}$$

$\mathbf{x}_C$ is a 12 × 1 state vector of the camera including the three-dimensional vectors of position $\boldsymbol{r}$, rotational angle φ, linear velocity $\mathbf{v}$, and angular velocity ω, all in world frame; $\mathbf{m}_i$ is the three-dimensional (3D) coordinates of ith stationary landmark in world frame; $\mathbf{O}_j$ is the state vector of jth moving object; n and l are the number of the landmarks and of the moving objects, respectively.

Figure 1 A free-moving monocular vision sensor.

The motion of the hand-held camera is presumed to be at constant velocity (CV), and the acceleration is caused by an impulse noise from the external force. Therefore, the state $\mathbf{x}_C$ of the camera with a CV motion model at time step k is expressed as:

$$\mathbf{x}_{Ck} = \begin{bmatrix} \mathbf{r}_k \\ \phi_k \\ \mathbf{v}_k \\ \boldsymbol{\omega}_k \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{k-1} + (\mathbf{v}_{k-1} + \mathbf{w}_{\mathbf{v}k-1})\Delta t \\ \phi_{k-1} + (\boldsymbol{\omega}_{k-1} + \mathbf{w}_{\omega k-1})\Delta t \\ \mathbf{v}_{k-1} + \mathbf{w}_{\mathbf{v}k-1} \\ \boldsymbol{\omega}_{k-1} + \mathbf{w}_{\omega k-1} \end{bmatrix} \tag{5}$$

where $\boldsymbol{w}_{\mathbf{v}}$ and $\boldsymbol{w}_{\omega}$ are linear and angular velocity noise caused by acceleration, respectively. The state of ith stationary landmark at time step k is represented by 3D coordinates in space,

$$\mathbf{m}_{ik} = [X_{ik}\ Y_{ik}\ Z_{ik}]^T \tag{6}$$

In the motion model of MOT, the targets to be tracked include the state and motion mode of the moving object in the environment. The state of the MOT system at time step k can be expressed as

$$\mathbf{O}_{jk} = [\boldsymbol{o}_{jk}\, \boldsymbol{s}_{jk}]^T, \text{ for } j = 1, 2, \ldots, l$$

where $\boldsymbol{o}_{jk}$ and $\boldsymbol{s}_{jk}$ are the state and motion mode of jth moving object, respectively. The MOT problem can be expressed as a probability density function (pdf) in Bayesian probability

$$p(\boldsymbol{o}_k, \boldsymbol{s}_k \mid \mathbf{z}_{1:k}) = p(\boldsymbol{o}_k \mid \boldsymbol{s}_k, \mathbf{z}_{1:k}) \cdot p(\boldsymbol{s}_k \mid \mathbf{z}_{1:k}) \tag{7}$$

where $p(\boldsymbol{o}_k|\boldsymbol{s}_k,\mathbf{z}_{1:k})$ is state inference; $\mathbf{z}_{1:k}$ is the set of measurements for time $t = 1$ to k; $p(\boldsymbol{s}_k|\mathbf{z}_{1:k})$ is the mode learning. The EKF-based interacting multiple model (IMM) estimator [13] can be utilized to estimate the motion mode of a moving object. The state is computed at time step k under each possible current model using r filters, with each filter using a different combination of the previous model-conditioned estimates. The mode M^j

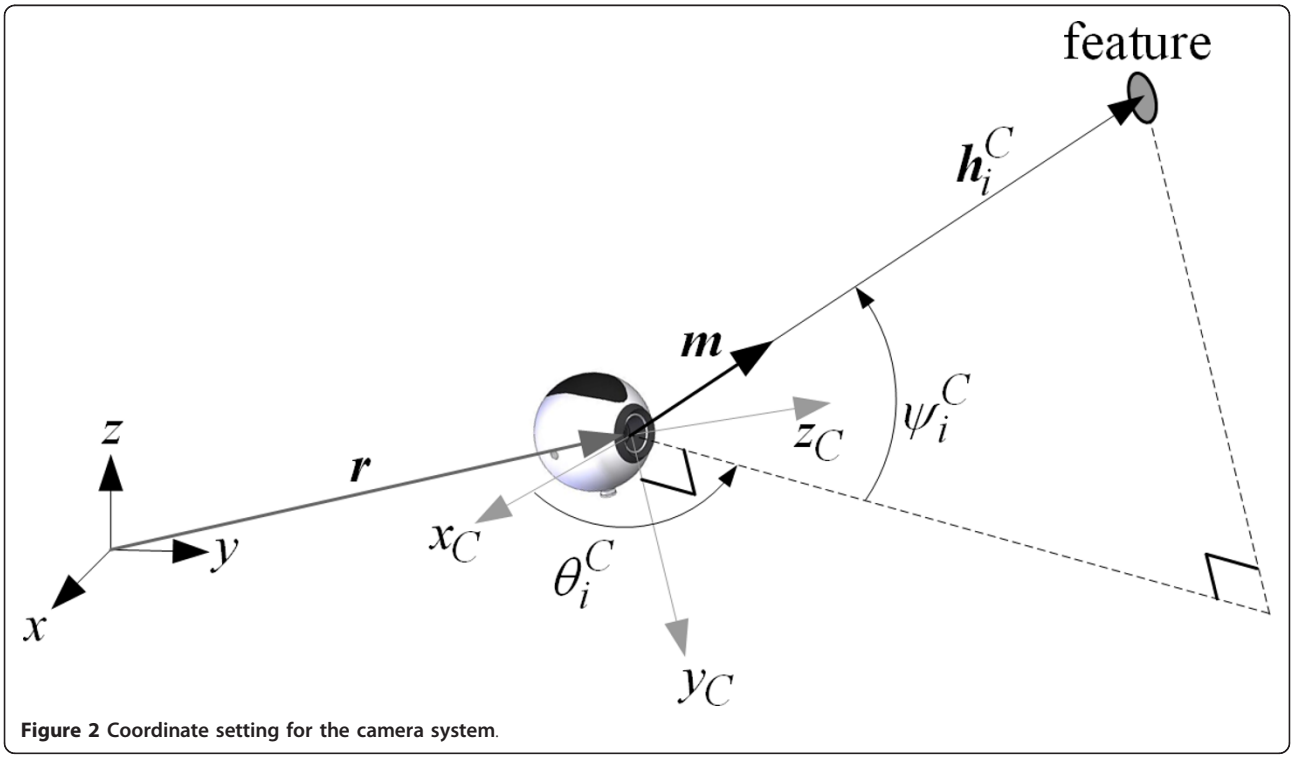

Figure 2 Coordinate setting for the camera system.

at time step k is assumed to be among the possible r modes

$$M^j \in \boldsymbol{M} = \left\{M^1,\ M^2,\ \cdots,\ M^r\right\}$$

Given r motion models, the object state $\boldsymbol{o}_k$ in Equation (7) is estimated. Instead of using the IMM estimator, a single CV model is utilized in the paper. That is, the moving objects are also presumed to move at a CV motion model. Their coordinates in 3D space are defined as

$$\boldsymbol{o}_{jk} = \begin{bmatrix} \boldsymbol{p}_{jk-1} + (\mathbf{v}_{jk-1} + \boldsymbol{w}_{jk-1})\Delta t \\ \mathbf{v}_{jk-1} + \boldsymbol{w}_{jk-1} \end{bmatrix} \tag{8}$$

where $\boldsymbol{p}_{jk}$ and $\mathbf{v}_{jk}$ are the vectors of the position and linear velocity of jth moving object at time step k, respectively.

2.2. Vision sensor model

The measurement vector of the monocular vision system is expressed as

$$\mathbf{z}_k = \left[\mathbf{z}_{1k}\ \mathbf{z}_{2k}\ \cdots\ \mathbf{z}_{mk}\right]^T$$

m is the number of the observed image features in current measurement. The perspective projection method [14] is employed to model the transformation from 3D space coordinate system to 2D image plane. For one observed image feature, the measurement is denoted as

$$\mathbf{z}_{ik} = \begin{bmatrix} I_{ix} \\ I_{iy} \end{bmatrix} = \begin{bmatrix} u_0 + f_u \dfrac{h^C_{ix}}{h^C_{iz}} + \alpha_C f_u \dfrac{h^C_{iy}}{h^C_{iz}} \\ v_0 + f_v \dfrac{h^C_{iy}}{h^C_{iz}} \end{bmatrix} \text{ for } i = 1,\ 2,\ \ldots,\ m \tag{9}$$

where f_u and f_v are the focal lengths of the camera denoting the distance from the camera center to the image plane in u- and v-axis, respectively; (u_0, v_0) is the offset pixel vector of the pixel image plane; α_C is the camera skew coefficient; $\boldsymbol{h}^C_i = \left[h^C_{ix}\ h^C_{iy}\ h^C_{iz}\right]^T$ is the ray vector of ith image feature in camera frame. The 3D coordinates of ith image feature or landmark in world frame, as shown in Figure 2, is given as

$$\mathbf{m}_i = \left[X_i\ Y_i\ Z_i\right]^T = \boldsymbol{r} + \boldsymbol{R}^W_C \boldsymbol{h}^C_i \tag{10}$$

$\boldsymbol{R}^W_C$ is the rotational matrix from world frame to camera frame, represented by using the elementary rotations [15],

$$\boldsymbol{R}^W_C = \begin{bmatrix} c\phi_y c\phi_z & s\phi_x s\phi_y c\phi_z - c\phi_x s\phi_z & c\phi_x s\phi_y c\phi_z + s\phi_x s\phi_z \\ c\phi_y s\phi_z & s\phi_x s\phi_y s\phi_z + c\phi_x c\phi_z & c\phi_x s\phi_y s\phi_z - s\phi_x c\phi_z \\ -s\phi_y & s\phi_x c\phi_y & c\phi_x c\phi_y \end{bmatrix} \tag{11}$$

where $c\varphi = \cos\varphi$ and $s\varphi = \sin\varphi$; φ_x, φ_y and φ_z are the corresponding rotational angles in world frame. We can utilize Equatoin (10) to calculate the ray vector of an image feature in camera frame. The coordinates of the feature in image plane are obtained by substituting Equations (10) and (11) into Equation (9) with $\alpha_C = 0$

$$I_{ix} = u_0 + f_u \frac{c\phi_y c\phi_z (X_i - r_x) + c\phi_y s\phi_z (Y_i - r_y) - s\phi_y (Z_i - r_z)}{(c\phi_x s\phi_y c\phi_z + s\phi_x s\phi_z)(X_i - r_x) + (c\phi_x s\phi_y s\phi_z - s\phi_x c\phi_z)(Y_i - r_y) + c\phi_x c\phi_y (Z_i - r_z)} \tag{12a}$$

$$I_{iy} = v_0 + f_v \frac{(s\phi_x s\phi_y c\phi_z - c\phi_x s\phi_z)(X_i - r_x) + (s\phi_x s\phi_y s\phi_z + c\phi_x c\phi_z)(Y_i - r_y) + s\phi_x c\phi_y (Z_i - r_z)}{(c\phi_x s\phi_y c\phi_z + s\phi_x s\phi_z)(X_i - r_x) + (c\phi_x s\phi_y s\phi_z - s\phi_x c\phi_z)(Y_i - r_y) + c\phi_x c\phi_y (Z_i - r_z)} \tag{12b}$$

Moreover, the elements of the Jacobian matrices $\boldsymbol{H}_k$ and $\boldsymbol{V}_k$ are determined by taking the derivative of $\mathbf{z}_i$ with respect to the state $\mathbf{x}_k$ and the measurement noise v_k. The Jacobian matrices are obtained for the purpose of calculating the innovation covariance matrix in EKF estimation process [16].

2.3. Feature initialization

Because of the lack of one-dimensional range information in image, how to initialize features becomes an important topic. Some researchers have successfully solved this problem either in time-delayed method [16] or un-delayed method [17]. The un-delayed method will be utilized in this research. When an image feature is selected, the spatial coordinates of the image feature are calculated by employing the method of inverse depth parameterization [17]. Assume that there are m image features with 3D position vectors, y_i, $i = 1,...,m$, which is described by the 6D state vector

$$\hat{y}_i = \left[\hat{r}^W_{ix}\ \hat{r}^W_{iy}\ \hat{r}^W_{iz}\ \hat{\theta}^W_i\ \hat{\psi}^W_i\ \hat{\rho}_i\right]^T \tag{13}$$

$\hat{r}^W = \left[\hat{r}^W_{ix}\ \hat{r}^W_{iy}\ \hat{r}^W_{iz}\right]^T$ indicates the estimated state of the camera when the feature was observed, as shown in Figure 2; $\hat{\rho}_i$ is the estimated image depth of the feature; $\hat{\theta}^W_i$ and $\hat{\psi}^W_i$ are the longitude and latitude angles of the spherical coordinate system which locates at the camera center. To compute the longitude and latitude angles, a normalized vector η^W_i in the direction of the ray vector is constructed by using the perspective project method:

$$\eta^W_i = R^W_C(\hat{\phi}^W)\left[\frac{I_{ix} - u_0}{f_u}\ \frac{I_{iy} - v_0}{f_v}\ 1\right]^T \tag{14}$$

Therefore, from Figure 2, the longitude and latitude angles of the spherical coordinate system can be obtained as

$$\hat{\theta}^W_i = \tan^{-1}\left(\frac{\eta^W_{iz}}{\eta^W_{ix}}\right) \tag{15a}$$

$$\hat{\psi}_i^W = \tan^{-1}\left(\frac{\eta_y^W}{\sqrt{\eta_{ix}^W 2 + \eta_{iz}^W 2}}\right) \tag{15b}$$

When image features are selected to be new landmarks or moving objects, the inverse depth parameterization vector in Equation (13) is assigned to be new augmented states in the EKF-based SLAM. Meanwhile, for each new state variable y_i, the corresponding covariance matrix are initialized according to

$$P_{k|k}^{new} = J\begin{bmatrix} P_{k|k} & 0 & 0 \\ 0 & R_i & 0 \\ 0 & 0 & \sigma_\rho^2 \end{bmatrix} J^T \tag{16}$$

$$J = \left[\begin{array}{c|c} I & 0 \\ \hline \frac{\partial \gamma}{\partial r^W} \; \frac{\partial \gamma}{\partial \phi^C} \; 0 \cdots 0 & \frac{\partial \gamma}{\partial z_i} \; \frac{\partial \gamma}{\partial \rho_i} \end{array}\right] \tag{17}$$

where R_i is the covariance of the measurement noise; σ_ρ is the deviation of the estimated image depth. However, the inverse depth coordinates are 6D and computational costly. A switching criterion is established in reference [17] based on a linearity index L_d. If $L_d > L_{d0}$, the inverse depth parameterization in Equation (13) is utilized as the augmented states; where L_{d0} is the threshold value of the index defined in [17]. On the other hand, if $L_d \le L_{d0}$, then the state vector in Equation (10) is chosen and modified as

$$\hat{Y}_i = \begin{bmatrix} \hat{Y}_{ix} \\ \hat{Y}_{iy} \\ \hat{Y}_{iz} \end{bmatrix} = \begin{bmatrix} \hat{r}_{ix}^W \\ \hat{r}_{iy}^W \\ \hat{r}_{iz}^W \end{bmatrix} + \frac{1}{\hat{\rho}_i} m(\hat{\theta}_i^W, \hat{\psi}_i^W) \tag{18}$$

$$m(\hat{\theta}_i^W, \hat{\psi}_i^W) = \begin{bmatrix} \cos(\hat{\theta}_i^W)\cos(\hat{\psi}_i^W) \\ \sin(\hat{\psi}_i^W) \\ \sin(\hat{\theta}_i^W)\cos(\hat{\psi}_i^W) \end{bmatrix} \tag{19}$$

where $m(\hat{\theta}_i^W, \hat{\psi}_i^W)$ is the unit ray vector. In this case, the corresponding covariance matrix is rearranged as

$$P_{\text{new}} = JPJ^T \tag{20}$$

$$J = \begin{bmatrix} I & 0 & 0 \\ 0 & \frac{\partial Y_i}{\partial y_i} & 0 \\ 0 & 0 & I \end{bmatrix} \tag{21}$$

Furthermore, for each new state variable υ_i, the correspondent elements of the Jacobian matrix H_k are modified as:

$$H_k = \frac{\partial g}{\partial x}(x_k, \upsilon_k) = \begin{bmatrix} \frac{\partial z_1}{\partial r^W} & \frac{\partial z_1}{\partial \phi^C} & \frac{\partial z_1}{\partial \mathbf{v}^W} & \frac{\partial z_1}{\partial \omega^C} & \frac{\partial z_1}{\partial \upsilon_1} & \cdots & \frac{\partial z_1}{\partial \upsilon_n} \\ \frac{\partial z_2}{\partial r^W} & \frac{\partial z_2}{\partial \phi^C} & \frac{\partial z_2}{\partial \mathbf{v}^W} & \frac{\partial z_2}{\partial \omega^C} & \frac{\partial z_2}{\partial \upsilon_1} & \cdots & \frac{\partial z_2}{\partial \upsilon_n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial z_m}{\partial r^W} & \frac{\partial z_m}{\partial \phi^C} & \frac{\partial z_m}{\partial \mathbf{v}^W} & \frac{\partial z_m}{\partial \omega^C} & \frac{\partial z_m}{\partial \upsilon_1} & \cdots & \frac{\partial z_m}{\partial \upsilon_n} \end{bmatrix} \tag{22}$$

The derivative is taken at $x_k = \hat{x}_{k|k-1}$ and $\nu_k = 0$.

2.4. Speeded-up robust features (SURF)

The basic concept of a scale-invariant method is to detect image features by investigating the determinant of Hessian matrix $\boldsymbol{H}$ in scale space [18]. In order to speed up the detection of image features, Bay et al. [10] utilize integral images and box filters to process on the image instead of calculating the Hessian matrix, and then the determinant of Hessian matrix is approximated by

$$\det(H)_{\text{approx.}} = D_{xx}D_{yy} - (wD_{xy})^2 \tag{23}$$

where D_{ij} are the images filtered by the corresponding box filters; w is a weight constant. The interest points or features are extracted by examining the extreme value of determinant of Hessian matrix. Furthermore, the unique properties of the extracted SURF are described by using a 64-dimensional description vector as shown in Figure 3[9,19].

2.5. Implementation of SLAM

The SLAM is implemented on the free-moving vision system by integrating the motion and sensor models, as well as the extraction of SURF. A flowchart for the developed SLAM system is depicted in Figure 4. The images are captured by the monocular camera and features are extracted using SURF method. In the SLAM flowchart, data association in between the landmarks in the database and the image features of the extracted SURF is carried out using the NN ratio matching strategy [9]. A map managerial tactic is designed to manage the newly extracted features and the bad features in the system. The details of the map management have been explained in previous article [19]. The properties of the newly extracted features are investigated and the moving objects will be discriminated from the stationary objects by using a proposed detection algorithm which will be described in next section. All the stationary landmarks and moving objects are included in the state vector. On the other hand, those features which are not continuously detected at each time step will be treated as bad features and erased from the state vector in Equation (4).

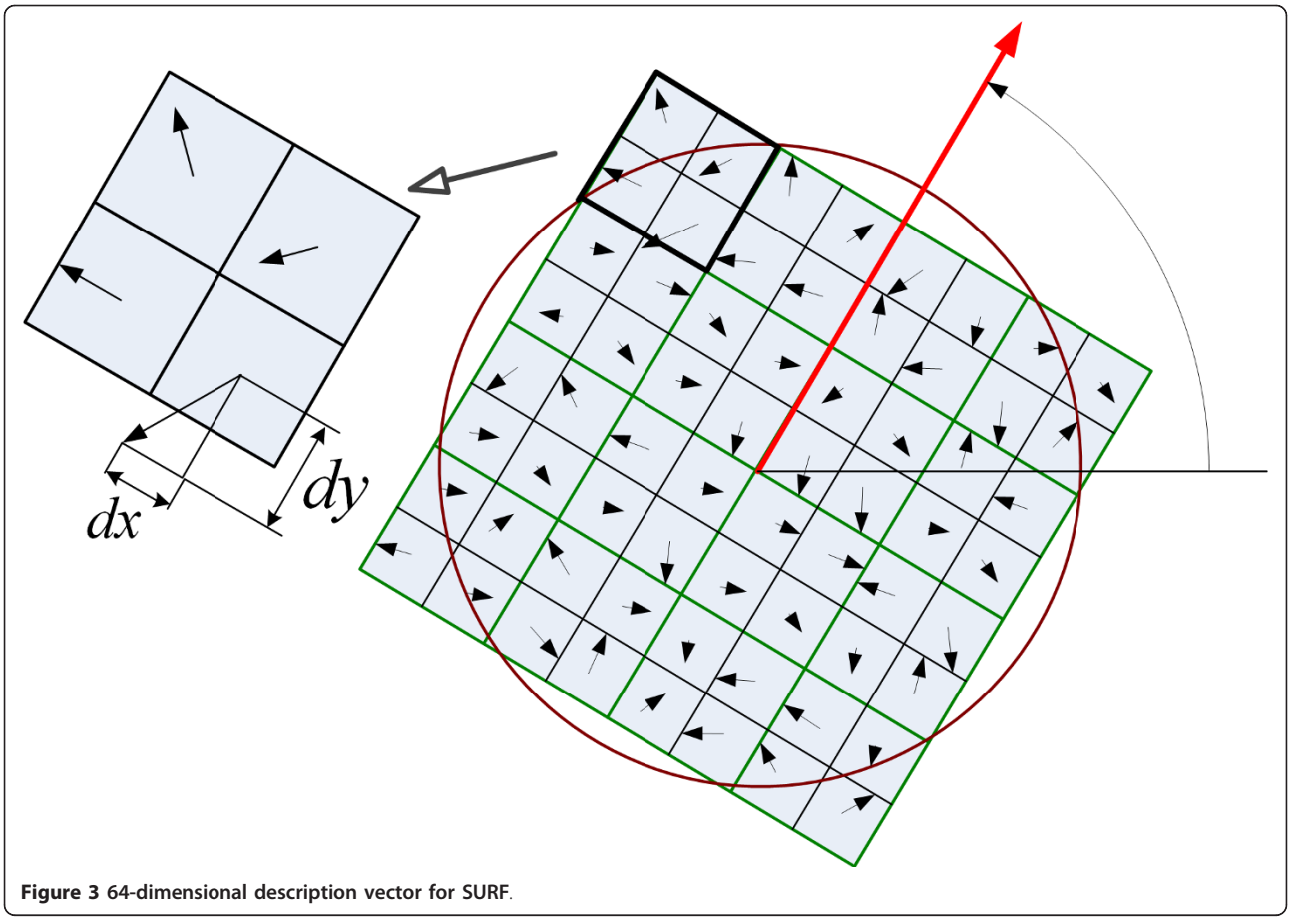

Figure 3 64-dimensional description vector for SURF.

3. Moving object detection and tracking

In the flowchart of Figure 4, the function block of MOD is designed based on the concept of the pixel coordinate constraint of a static object in image plane. An object in space is represented by the corresponding features in two consecutive images. The pixel coordinate constraint equation for these corresponding image features can be expressed as

$$\boldsymbol{h}_d^{CT} \boldsymbol{E} \boldsymbol{h}'^{C}_{d} = 0 \tag{24}$$

where $\boldsymbol{h}_d^C$ and $\boldsymbol{h}'^{C}_{d}$ are the homogenous normalized image coordinates of the corresponding features abstracted from two consecutive images, images 1 and 2, respectively. They are defined as

$$\boldsymbol{h}_d^C = \boldsymbol{K}_C^{-1} \begin{bmatrix} I_x \\ I_y \\ 1 \end{bmatrix} \boldsymbol{h}'^{C}_{d} = \boldsymbol{K}_C^{-1} \begin{bmatrix} I'_x \\ I'_y \\ 1 \end{bmatrix}; \text{ and } \boldsymbol{K}_C = \begin{bmatrix} f_u & \alpha_C f_u & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$\boldsymbol{K}_C$ is the matrix of camera intrinsic parameters which can be obtained from Equation (9). Note that the image coordinates are defined in camera frame. The essential matrix $\boldsymbol{E}$ in Equation (24) is defined as [20]

$$\boldsymbol{E} = [\boldsymbol{t}]_\times \boldsymbol{R} = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \boldsymbol{R} \tag{25}$$

where $\boldsymbol{R}$ is the rotation matrix and $\boldsymbol{t}$ is the translation vector of the camera frame with respect to world frame; $[\boldsymbol{t}]_\times$ is the matrix representation of the cross product with $\boldsymbol{t}$. The rotation matrix and translation vector would be determined using EKF estimator. Therefore, the essential matrix $\boldsymbol{E}$ can be calculated accordingly. Usually, the pixel coordinate constraint in Equation (24) is utilized to estimate the state vector and according essential matrix. Given a set of corresponding image points, it is possible to estimate the state vector and the essential matrix which optimally satisfy the pixel coordinate constraint. The most straight-forward approach is to set up a total least squares problem, commonly known as the eight-point algorithm [21]. On the contrary in SLAM problem, the state vector and the essential matrix is obtained using the state estimator. We could further utilize the estimated state vector and the essential matrix to investigate whether a set of corresponding image points satisfy the pixel coordinate

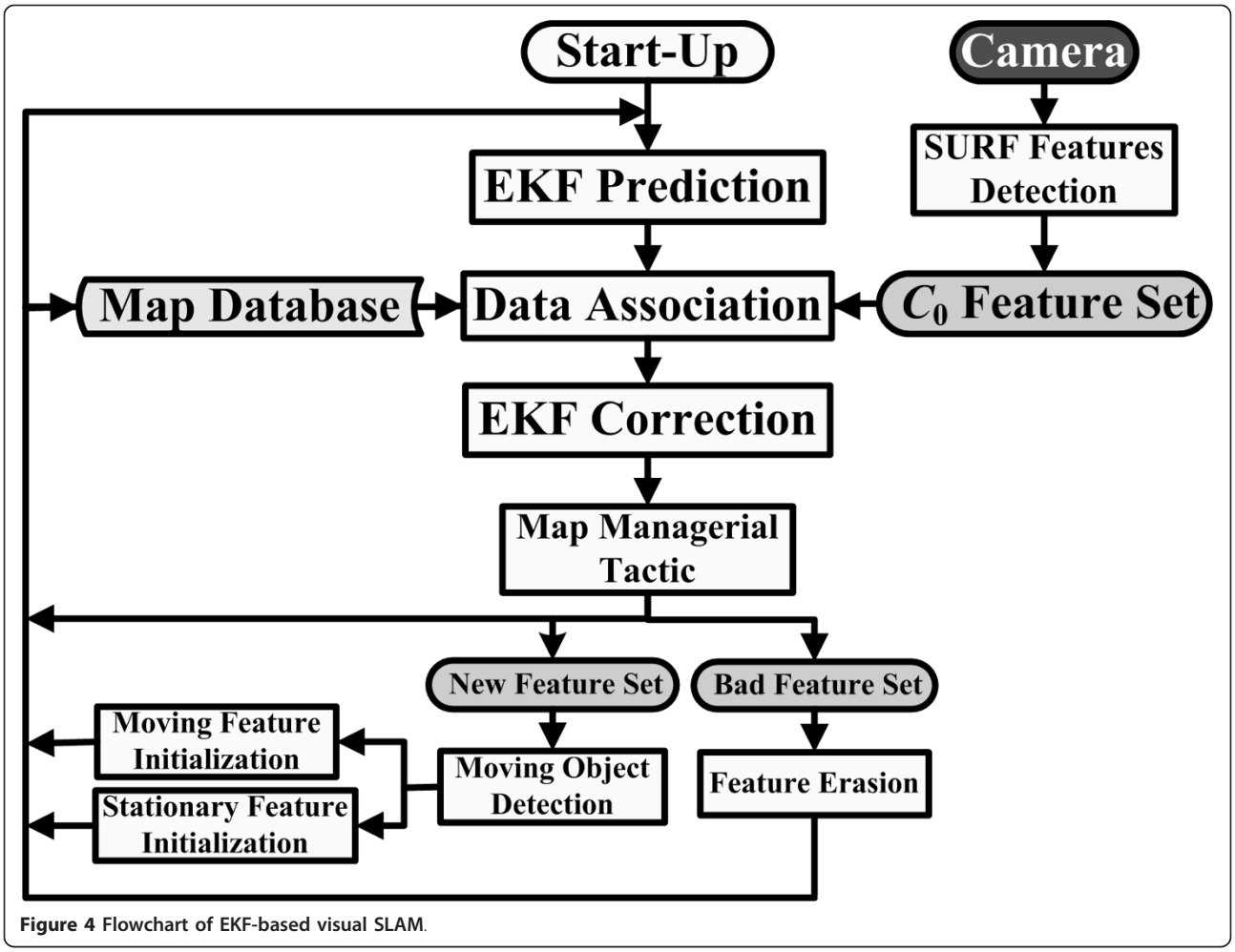

Figure 4 Flowchart of EKF-based visual SLAM.

constraint in Equation (24). First, define the known quantities in Equation (24) as a constant vector

$$\begin{bmatrix} a\ b\ c \end{bmatrix} = \begin{bmatrix} I_x\ I_y\ 1 \end{bmatrix}^T (\boldsymbol{K}_C^{-1})^T \boldsymbol{E} \boldsymbol{K}_C^{-1} \tag{26}$$

Equation (26) is further rearranged as an equation of the epipolar line constraint in image plane for two corresponding points [22], as shown in Figure 5

$$aI'_x + bI'_y + c = 0 \tag{27}$$

Equation (27) indicates that the pixel coordinate of the corresponding feature in second image will be constrained on the epipolar line.

Simulation of pixel constraint on the corresponding features for the cases of camera translation and rotation is carried out and depicted in Figure 6. Assume that the image feature of a static object in first image is located at $(I_x, I_y) = (80,60)$. In the first simulation, the camera translates 1 cm along x_c-axis direction, the corresponding image feature (I'_x, I'_y) in the second image must be restricted in the epipolar line 1, as shown in Figure 6. Similarly, if the camera translates 1 cm along y_c- or z_c-axis direction, the corresponding features (I'_x, I'_y) in the second image must be located on the epipolar line 2 or 3, respectively. In the second simulation, the camera first moves 1 cm in x_c-axis and then rotates $\varphi_x = 15°$ about x_c-axis, the corresponding feature (I'_x, I'_y) in second image must be located on the epipolar line 4, as shown in Figure 6. If the camera first moves 1 cm in x_c-axis and then rotates $\varphi_y = 15°$ about y_c-axis or $\varphi_z = 15°$ about z_c-axis, the corresponding features (I'_x, I'_y) in the second image must be located on the epipolar line 5 or 6, respectively.

Motion and measurement noise is involved in the state estimation process. If the measurement of image points is subject to noise, which is the common case in any practical situation, it is not possible to find a corresponding feature point which satisfies the epipolar constraint exactly. That is, the corresponding feature point might not be constrained on the epipolar line because

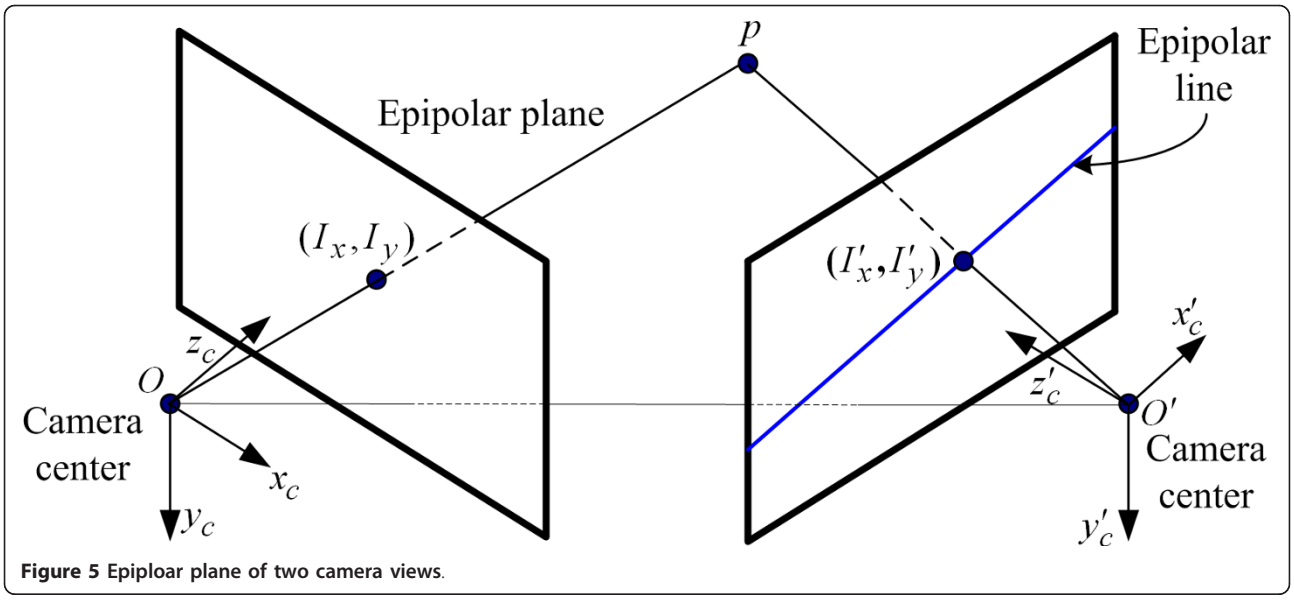

Figure 5 Epiploar plane of two camera views.

of the presence of noise. Define the distance D to represent the pixel deviation of the corresponding point from the epipolar line in image plane, as shown in Figure 7,

$$D = \left| \frac{aI'_x + bI'_y + c}{\sqrt{a^2 + b^2}} \right| \tag{28}$$

D is utilized in this article to denote the pixel deviation from the epipolar line which is induced by the motion and measurement noise in state estimation process. Depending on how the noise related to each constraint is measured, it is possible to design a threshold value in Equation (28) which satisfies the epipolar constraint for a given set of corresponding image points. For example, the image feature of a static object in the first image is located at (I_x, I_y) = (50,70) and then the camera moves 1 cm in z_c-axis. If the corresponding image feature (I'_x, I'_y) in the second image is constrained within a deviation I'_y is limited in a range, as shown in Figure 7, as I'_x varying from 1 to 320.

There are two situations that the pixel coordinate constraint in Equation (24) will result in trivial solutions. First, if the camera is motionless between two consecutive images, we can see from Equation (25) that the essential matrix becomes a zero matrix. Therefore, the object status could not be obtained by investigating the pixel coordinate constraint. In this case, we assume the camera being stationary in space and compute the pixel distance in between the corresponding features in two consecutive images to determine the object status. Second, if the image feature of a static object in the first image is located near the center of image plane (I_x, I_y) = (160,120), the coefficients of the epipolar line obtained from Equation (26) are zero. Therefore, any point in the second image will satisfy the equation of the epipolar line in Equation (27). This situation is simulated and the result is depicted in Figure 8. In the simulation, the pixel coordinate in the first image I_x varies from 0 to 320 and I_y is fixed at 120. The corresponding feature (I'_x, I'_y) in the second image is limited within the pixel coordinate constraint $\boldsymbol{h}_d^{CT}\boldsymbol{E}\boldsymbol{h}'^{C}_{d} < 10^{-3}$. We can see that the range of the corresponding I'_y is unlimited when (I_x, I_y) is close to (160,120), as shown in Figure 8. In real applications, those features located in a small region near the center of the image plane need a special treatment because the epipolar line constraint is not valid in this situation.

4. Experimental results

In this section, the experimental works of the online SLAM with a moving object detector are implemented on a laptop computer running Microsoft Window XP. The laptop computer is Asus U5F with Intel Core 2 Duo T5500 (1.66 GHz), Mobile Intel i945GM chipset and 1 Gb DDR2. The free-moving monocular camera utilized in this work is Logitech C120 CMOS web-cam with 320 × 240-pixels resolution and USB 2.0 interface. The camera is calibrated using the Matlab tool provided by Bouquet [23]. The focal lengths are f_u = 364.4 pixels and f_v = 357.4 pixels. The offset pixels are u_0 = 156.0 pixels and v_0 = 112.1 pixels, respectively. We carried out three experiments including the SLAM task in a static environment, SLAM with MOT, and people detection and tracking.

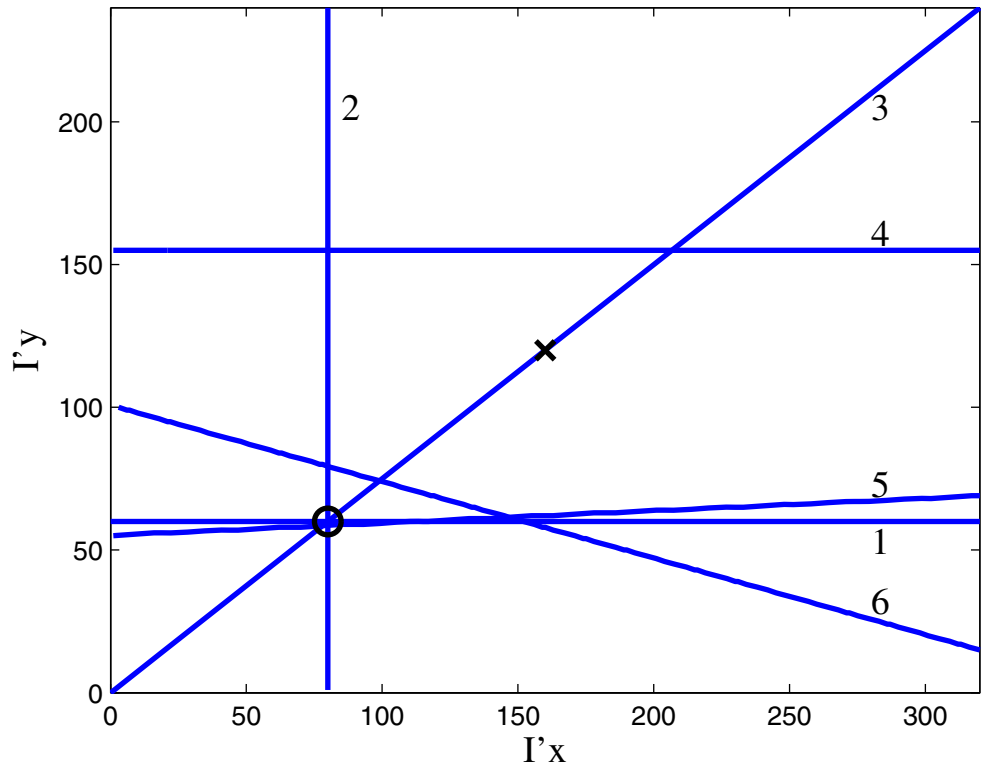

Figure 6 Epipolar lines.

4.1. SLAM in static environment

In this experiment, the camera is carried by a person to circle around a bookshelf (1.5 m × 2 m floor dim.) in our laboratory. The resultant map and the camera pose estimation are plotted in Figure 9. In this figure, the estimated states of the camera and landmarks are illustrated in a 3D plot. The ellipses in the figure indicate the uncertainty of the landmarks obtained from the extracted image features. The rectangular box represents the free-moving monocular camera and the solid line depicts the trajectory of the camera. More detail image frames of the experimental results are illustrated in Figures 10, 11, 12, 13, 14, 15, 16, and 17. For each figure, the captured image is shown in the left panel and the top-view plot is depicted in the right panel. The (blue) circular marks in the left panel of the figures indicate the landmarks extracted from the captured image with an unknown image depth, while the (red) square marks represent the landmarks with a known and stable image depth. In the right panel of the figures, the estimated states of the camera and landmarks are illustrated in a 2D plot. The red (dark) ellipses represent the uncertainty of the landmarks which have known image depths and the green (light) ellipses denote the uncertainty of the landmarks which have unknown image depths. Meanwhile, the rectangular box represents the free-moving camera and the trajectory of the estimated camera pose is plotted as solid lines. As shown in the right panel of Figure 10 for the 31st image frame, the SLAM system starts up and captures four image features with known positions. These features will help to initialize the map scale. After the start-up, more image features are extracted and treated as landmarks with unknown image depth, as indicated in circular marks in Figure 11 for the 32nd frame. In Figure 12 for the 71st frame, the uncertainty of the image depth of feature 10 is reduced to a small region (red ellipse). The SLAM system builds the environment map and estimates the camera pose concurrently, when the camera is carried to circle around the laboratory, as shown in Figure 13 for the 645th frame. In Figures 14 and 15 for the 2265th and 2340th frames, the camera comes to the place it had visited before and the trajectory loop is closing. Some old landmarks (with number less than 100) are captured again and the covariance of the state vector is reduced gradually. The second and third times of loop-closure are depicted in Figures 16 and 17 for the 3355th and 4254th frames. In these frames, old landmarks are

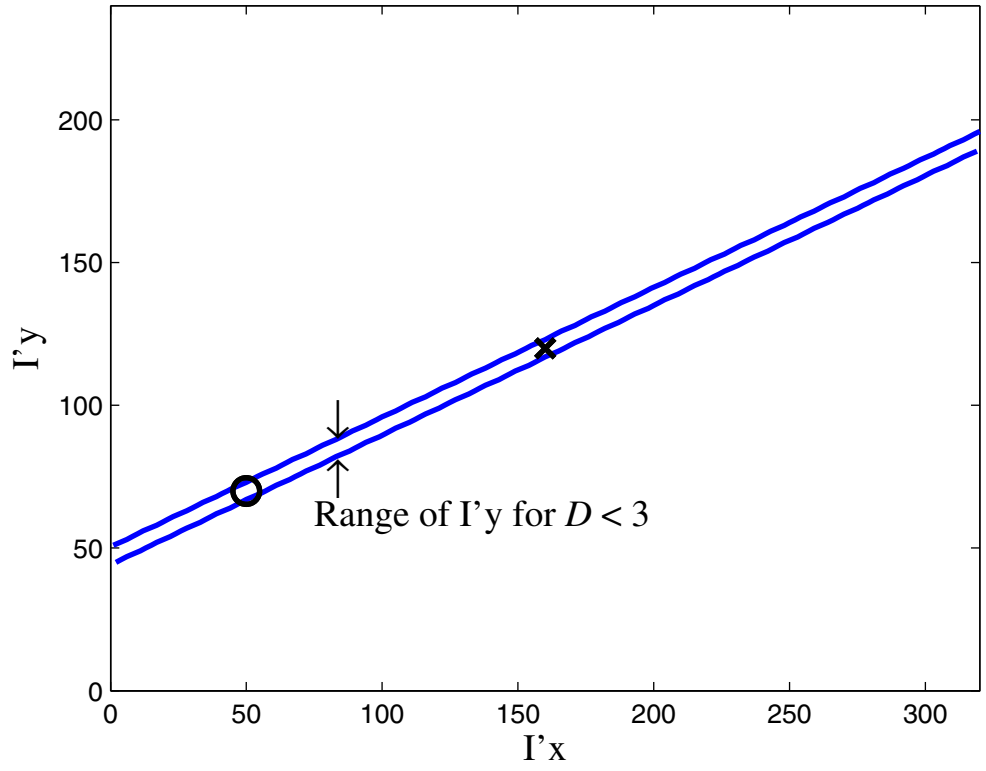

Figure 7 Deviation from the epipolar line.

visited again and the covariance of the state vector is reduced further.

The map size and the sampling frequency (Hz) at each frame are depicted in Figure 18. The map size increases to about 145 features at the first loop-closure and remains at almost constant size (about 200 features at the third loop-closure). That is, the SLAM can rely on the map with old landmarks for localization when it repeats visiting the same environment. The high sampling frequency in Figure 18 is about 20 Hz when the map size is small. When the map size increases, the low sampling frequency keeps at about 5 Hz.

The deviations of the camera pose in *xyz*-axis are plotted in Figure 19. The camera pose deviations decrease suddenly at each loop-closure, because the old landmarks are revisited and the camera pose and landmark locations are updated accordingly. We also can see from Figure 19 that the lower-bound of the pose deviation is further decreased after each loop-closure.

4.2. SLAM with MOT

The camera is carried to move around at one corner of our laboratory in this example. Meanwhile, the SLAM is implemented to map the environment and estimate the camera pose, as well as to detect and track a moving object. The estimated camera pose and landmarks are illustrated in a 3D plot as shown in Figure 20. The ellipses in the figure indicate the landmarks obtained from the extracted image features. The rectangular box represents the free-moving monocular camera and the solid line depicts the trajectory of the camera. More detail image frames of experimental results are illustrated in Figures 21, 22, 23, 24, 25, 26, 27, 28, 29 and 30. For each figure, the captured image is shown in the left panel and the top-view plot is depicted in the right panel of each figure. As shown in Figure 21 for the 52nd image frame, the SLAM system starts up and captures four image features with known 3D coordinates. After the start-up, some stationary features with unknown status are extracted and treated as new landmarks for mapping, as shown in the 98th frame in Figure 22. These stationary landmarks are initialized using inverse depth parameterization [17]. The system demonstrates a stable implementation of SLAM as shown in Figures 22, 23, 24, and 25. In Figure 26, soon after the feature no. 30 is detected, it is discriminated from the stationary features using the proposed MOD algorithm and then treated as a moving object. This moving object

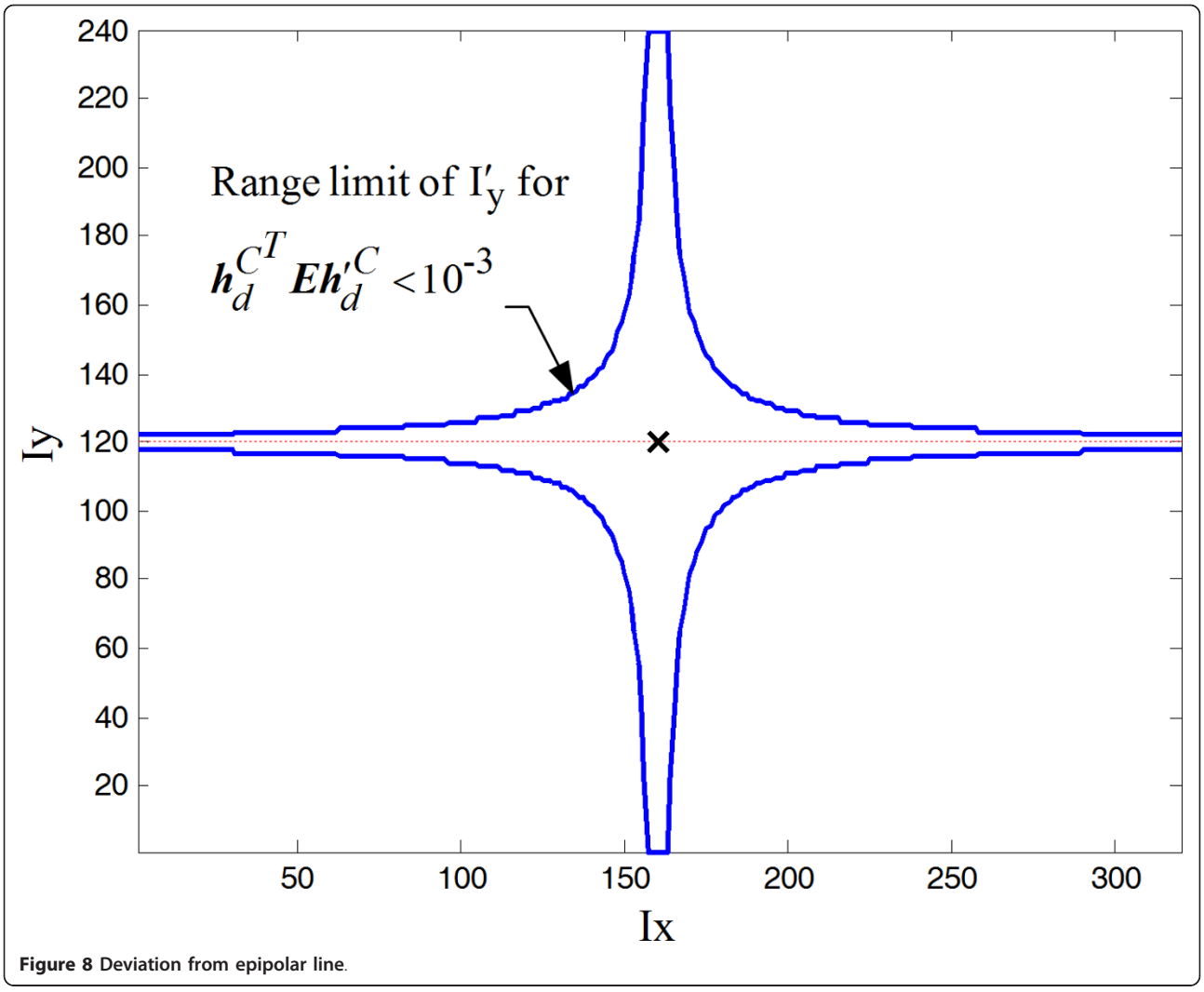

Figure 8 Deviation from epipolar line.

is tracked in the image plane and encircled by a rectangle in Figures 26, 27, 28, 29 and 30. Therefore, the developed system demonstrates the capability of simultaneous localization, mapping and MOT.

4.3. People detection and tracking

In this experiment, the SLAM system with MOD is utilized to detect and track a moving people in the environment. The camera is carried to move around at a corner of our laboratory. The estimated states of the camera pose, the landmarks, and the moving people are illustrated in a 2D top-view plot in the right panel of Figures 31, 32, 33, 34, and 35. The captured image is shown in the left panel of each figure. As shown in Figure 31 for the 350th image frame, the SLAM system builds the environment map and estimates the camera pose concurrently and stably. The ellipses in the right panel indicate the landmarks obtained from the extracted image features. The rectangular box represents the free-moving monocular camera and the thin solid line depicts the trajectory of the camera. One person gets in the scene, as shown in the 365th frame in Figure 32, and two image features on the human body are detected. These image features are discriminated from the stationary features using the proposed MOD algorithm, and are further initialized in state vector and tracked using Equation (8). The SLAM system continuously tracks the moving people until he goes out the scene, as shown in Figures 33, 34, and 35. The trajectory of the moving people is depicted as a thick solid line in the right panel of each figure. Hence, the developed system also demonstrates the capability of simultaneous localization, mapping, and moving people tracking.

5. Conclusions

In this research, we developed an algorithm for detection and tracking of moving objects to improve the robustness of robot visual SLAM system. SURFs are also

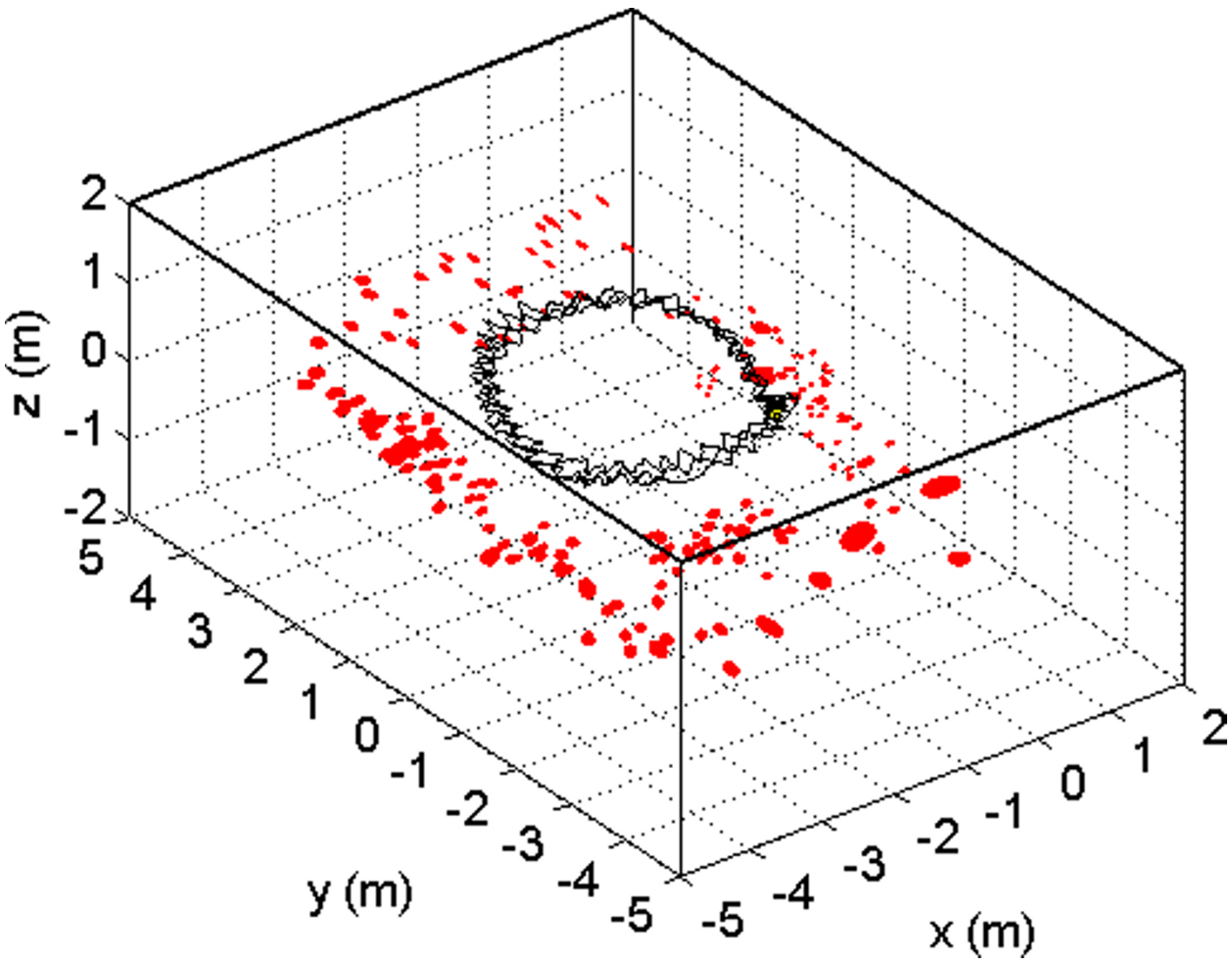

Figure 9 Three-dimensional map and camera pose estimation.

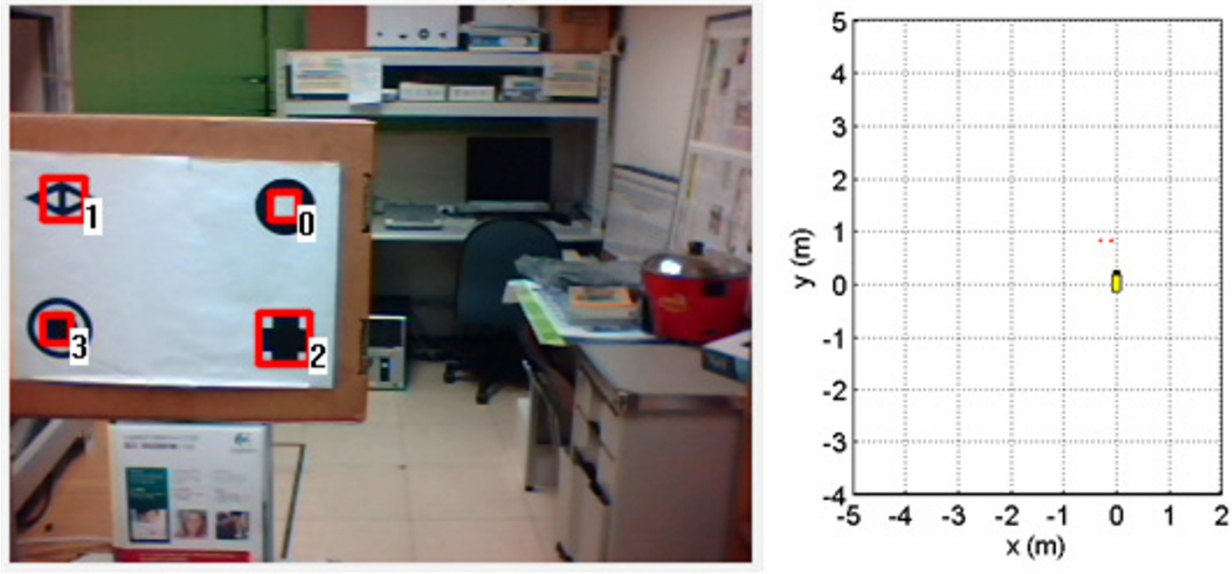

Figure 10 31st frame: the SLAM system starts up with four known features, 0, 1, 2, 3.

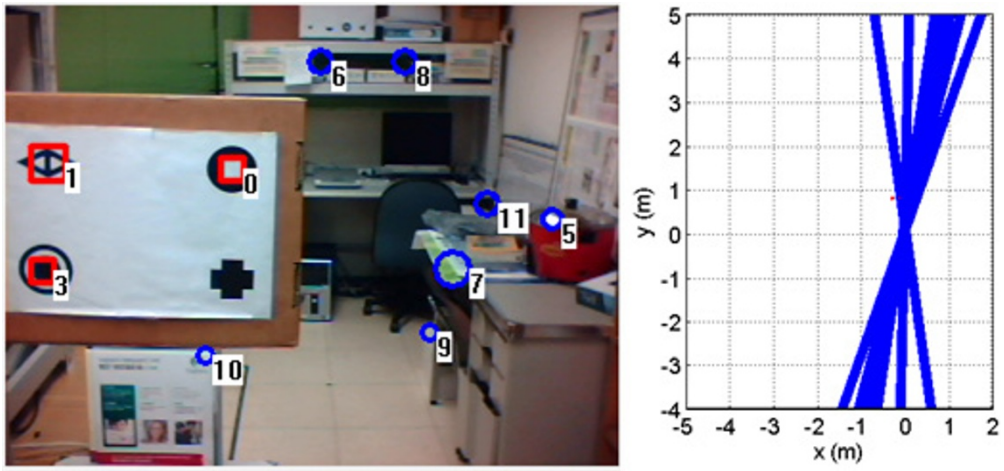

Figure 11 32nd frame: some new features are detected and initialized.

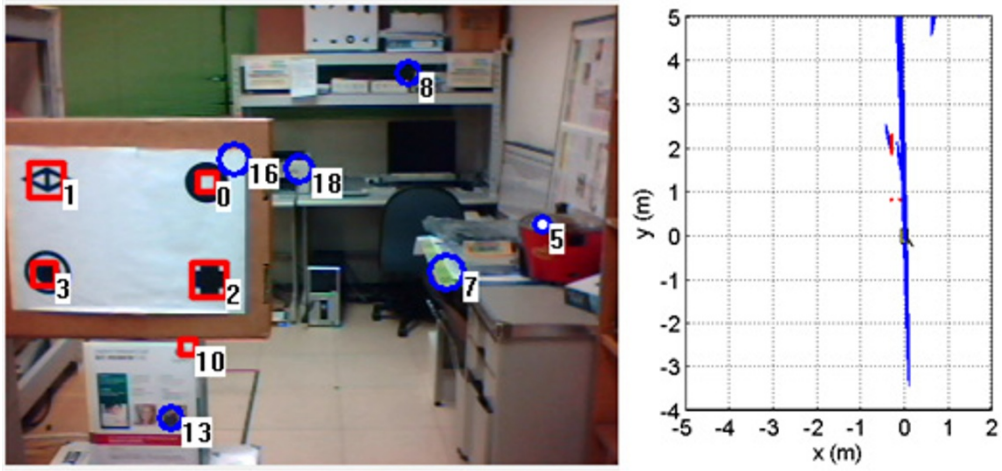

Figure 12 71st frame: the image depth uncertainty of feature 10 decreases to a small region.

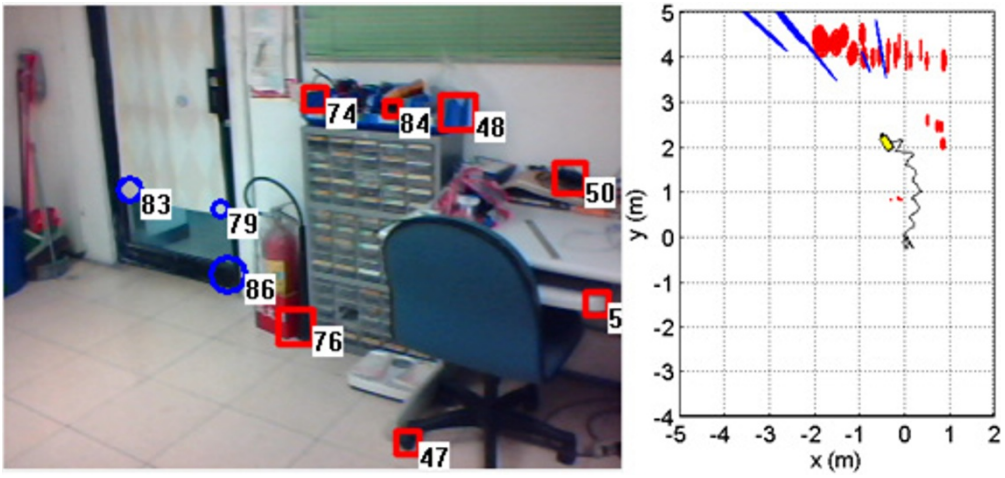

Figure 13 645th frame: the SLAM system maps the environment and estimates the camera pose concurrently.

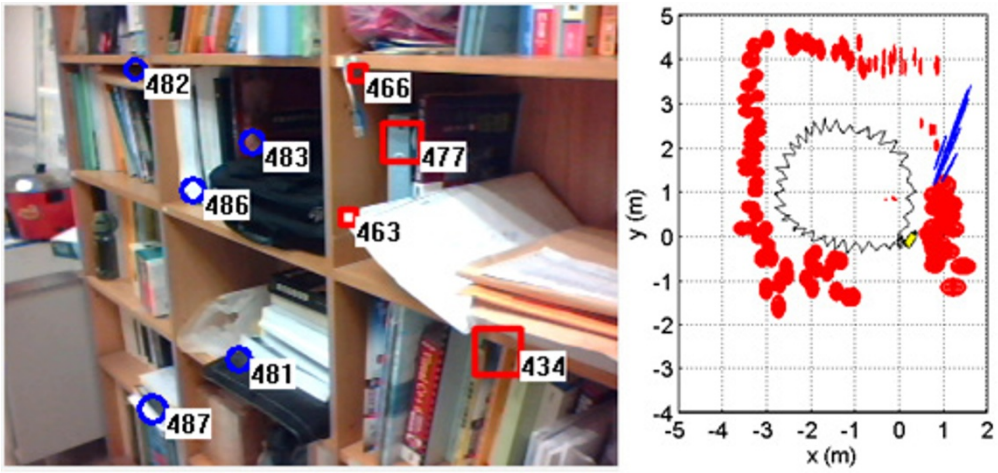

Figure 14 2265th frame: before the loop is closing.

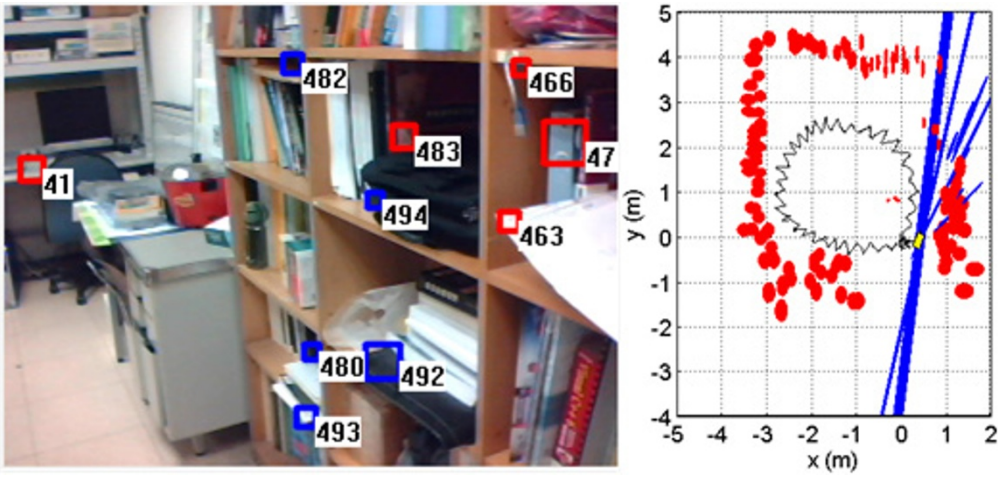

Figure 15 2340th frame: the first time the camera reaches the loop-closure.

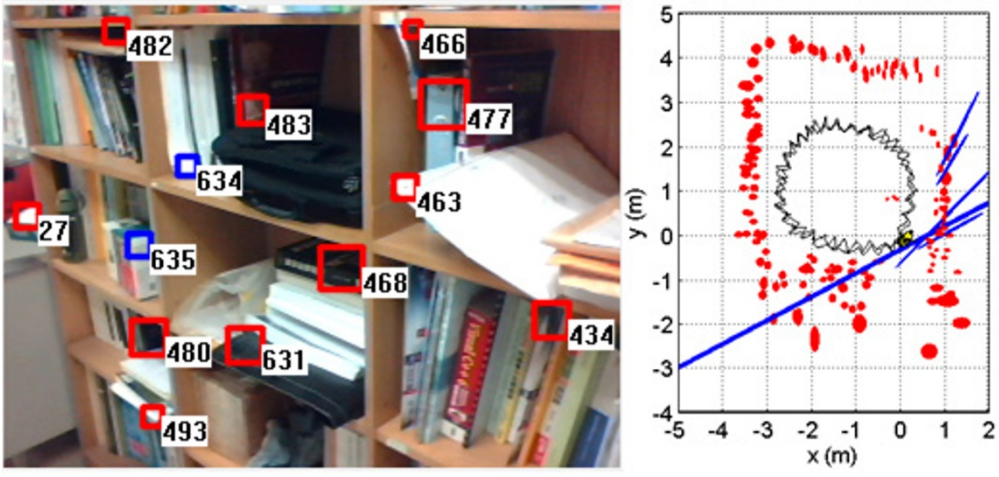

Figure 16 3355th frame: the second time the camera reaches the loop-closure.

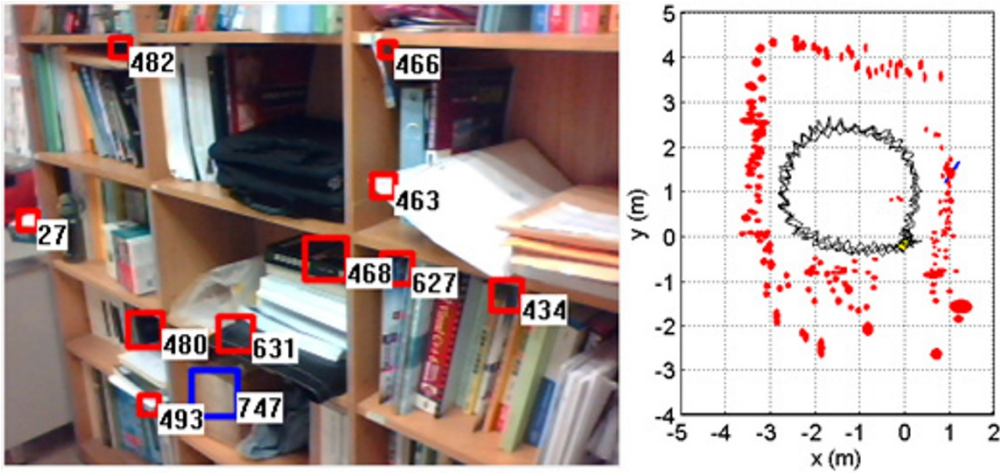

Figure 17 4254th frame: the third time the camera reaches the loop-closure.

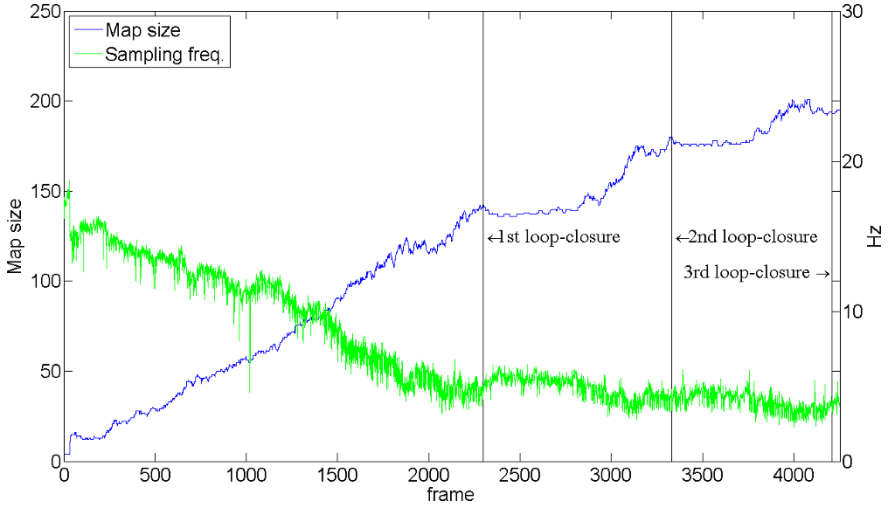

Figure 18 Map size and sampling frequency.

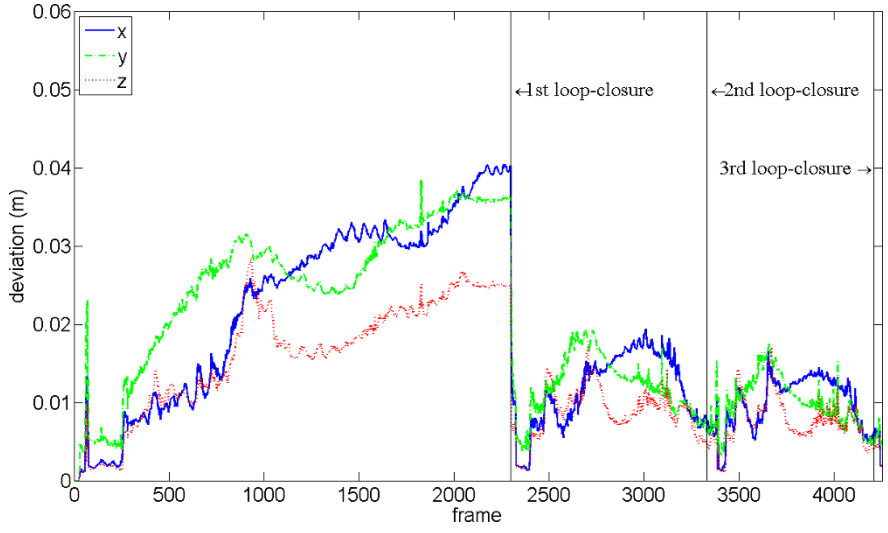

Figure 19 The deviation of the camera pose estimation.

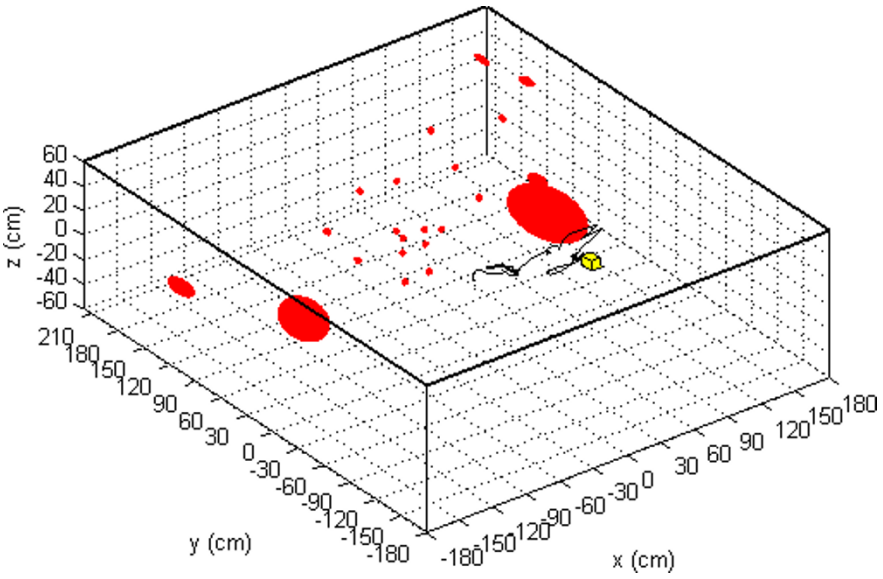

Figure 20 Three-dimensional map and camera pose estimation.

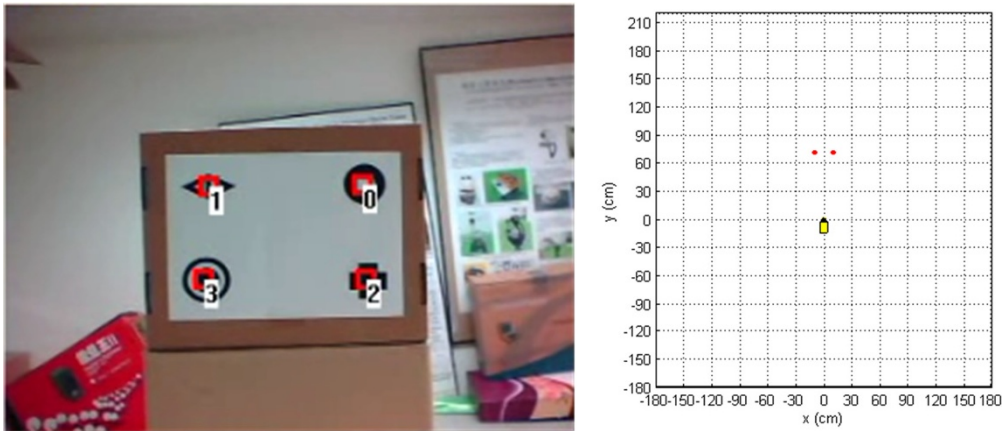

Figure 21 52nd frame: system start-up with four known features, 0, 1, 2, 3.

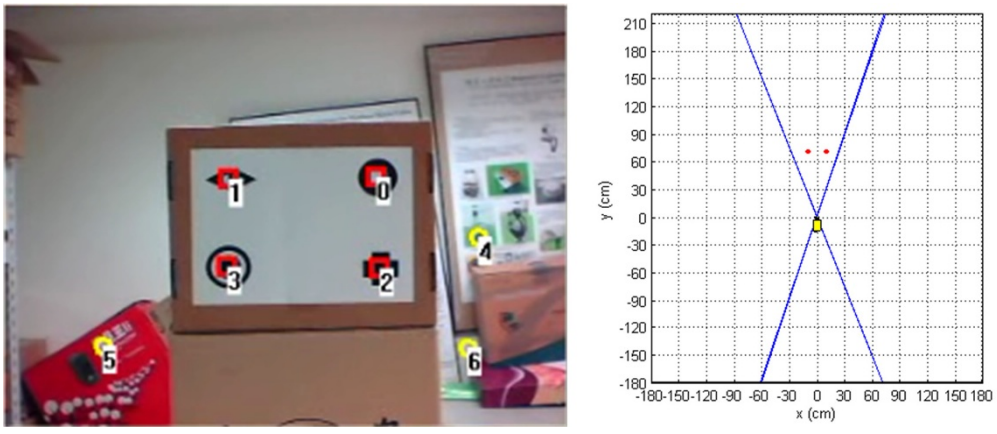

Figure 22 98th frame: three new features are initialized.

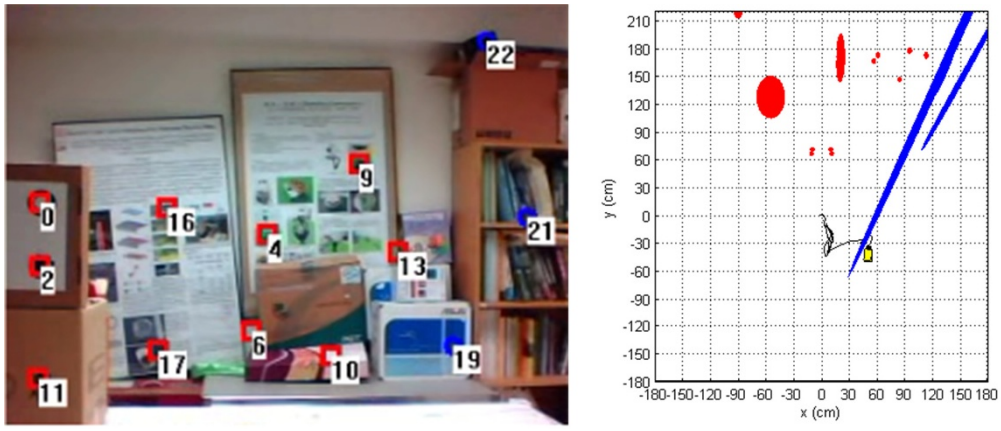

Figure 23 330th frame: SLAM.

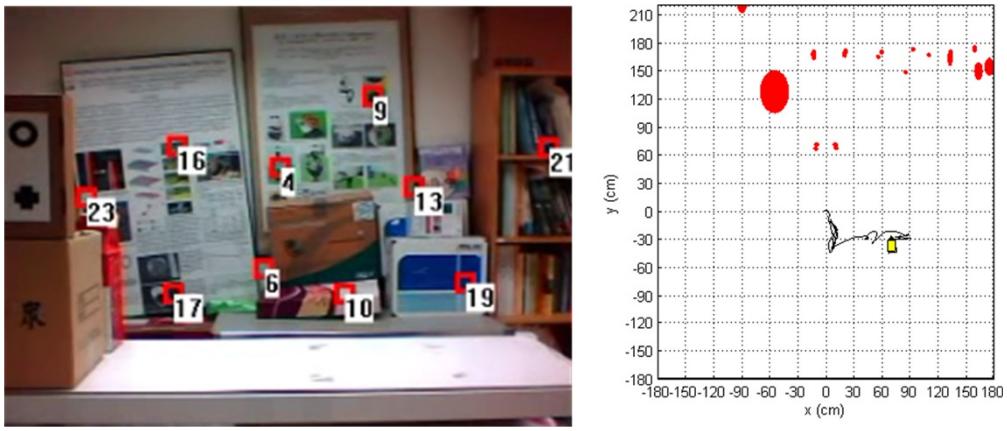

Figure 24 526th frame: SLAM.

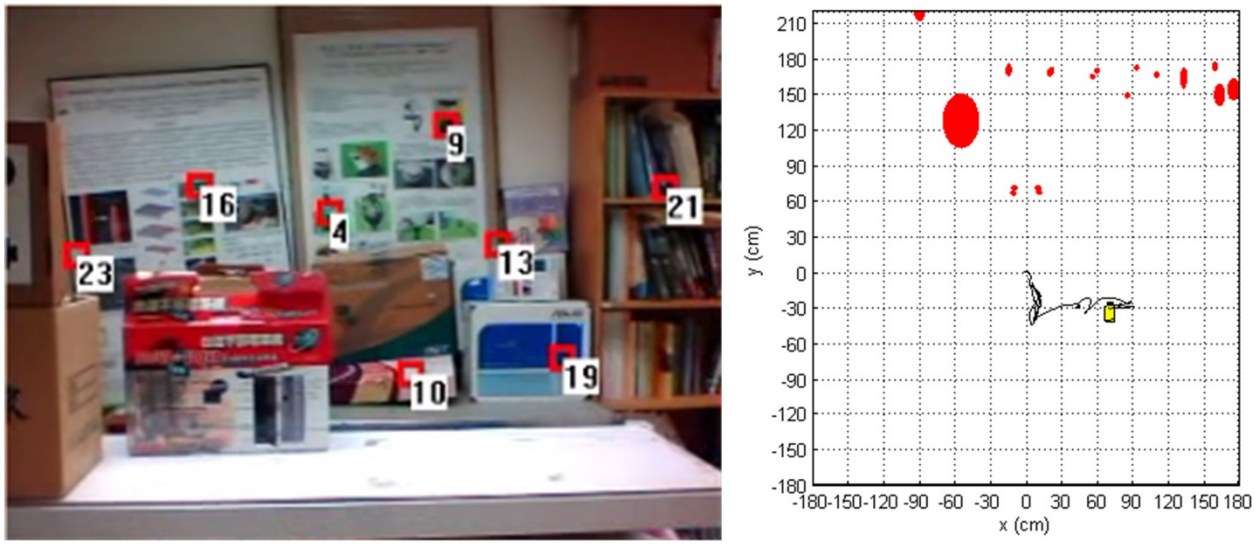

Figure 25 563rd frame: one object (the box) is moving to right direction.

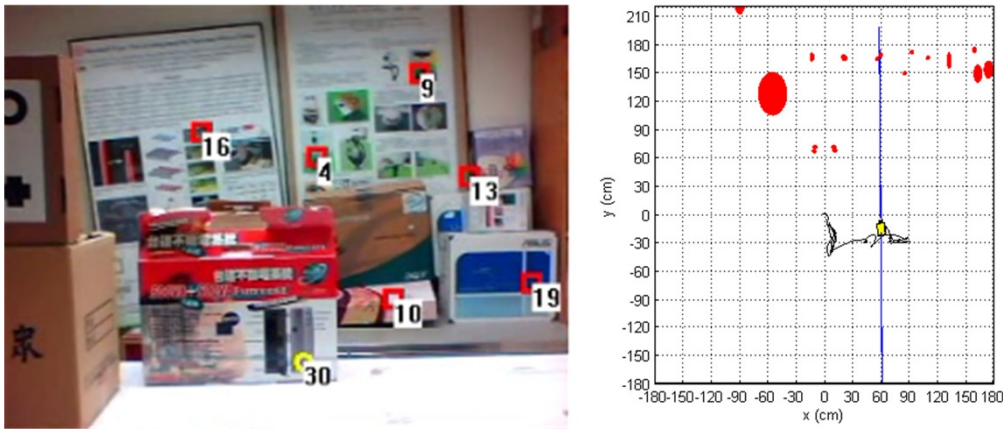

Figure 26 615th frame: feature 30 on the object is deteced and initialized with large image depth uncertainty.

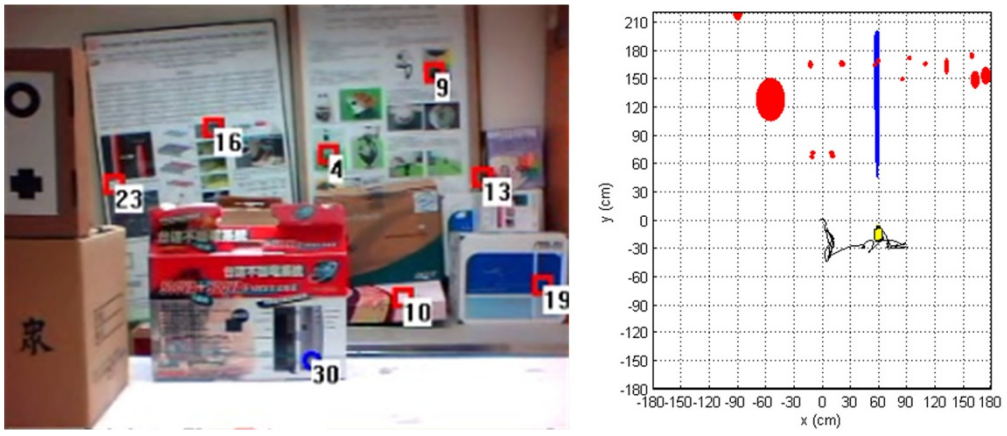

Figure 27 621st frame: the object stops moving and the image depth uncertainty of the feature is reduced.

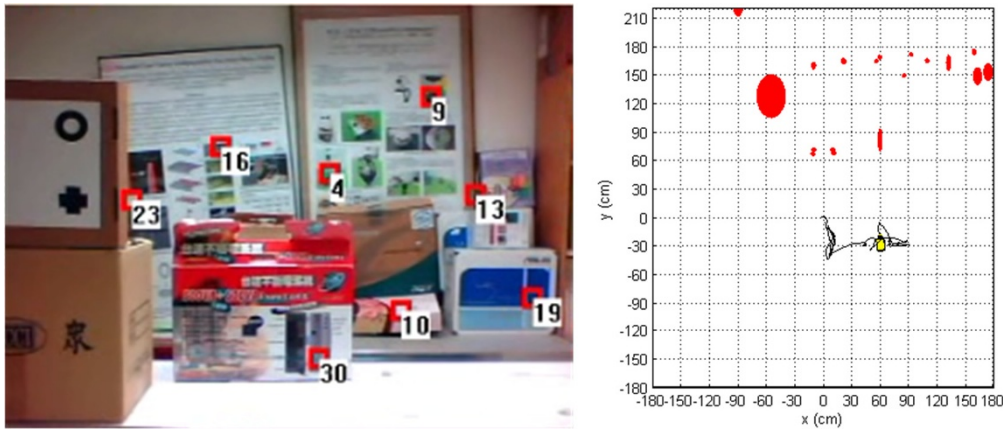

Figure 28 634th frame: the image depth uncertainty of the tracked feature is further reduced and treated as a 3D point.

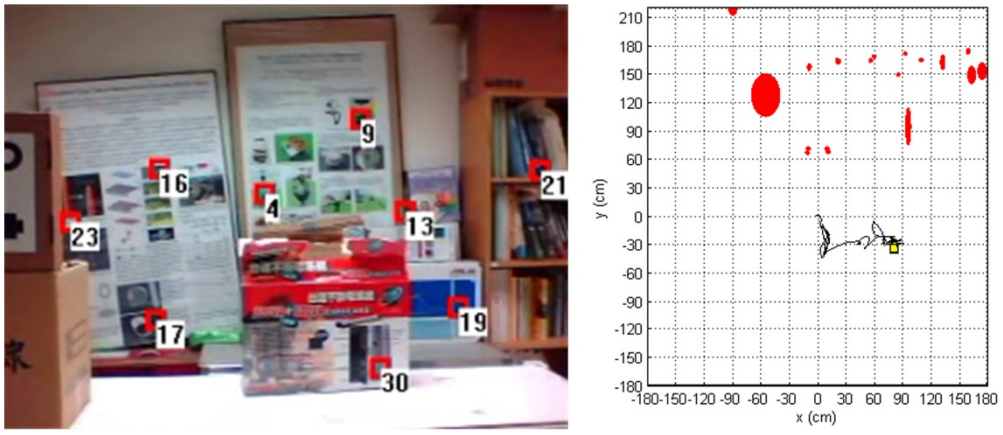

Figure 29 657th frame: the object moves to the right again and the covariance increases.

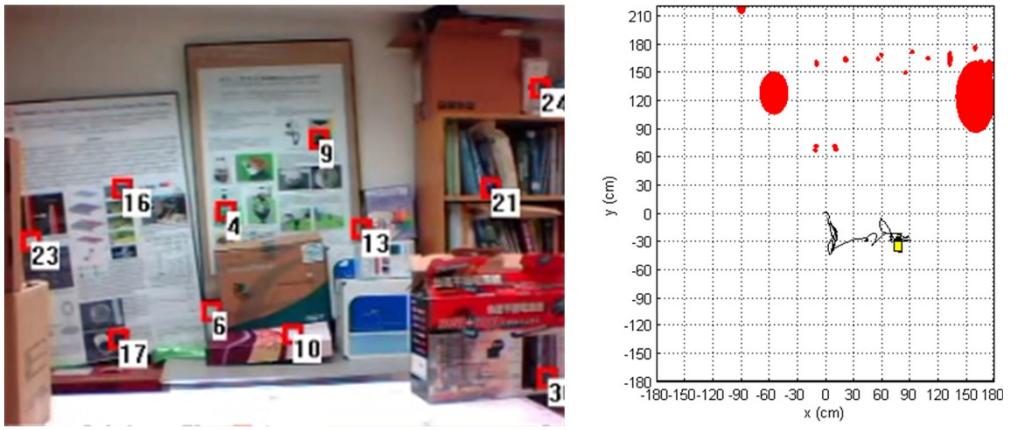

Figure 30 684th frame: the object keeps moving to the right and the covariance increases further.

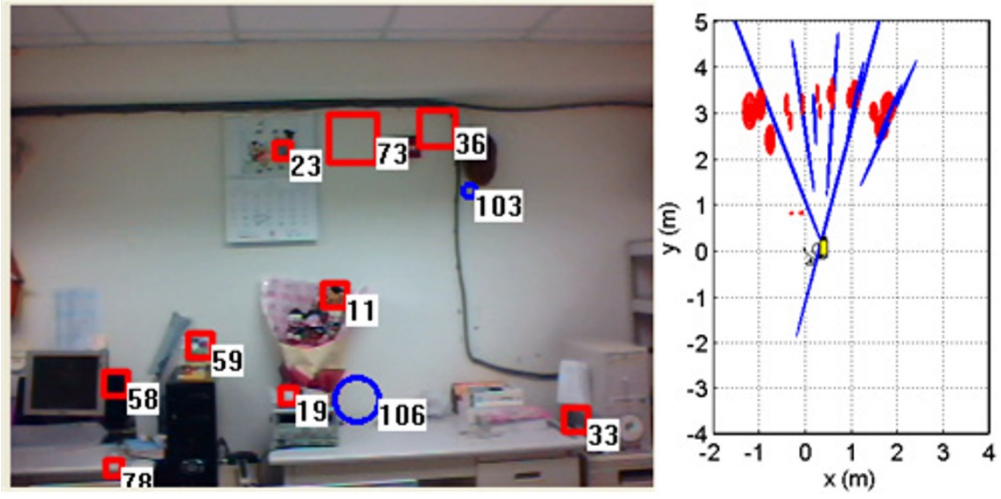

Figure 31 350th frame: the system performing SLAM.

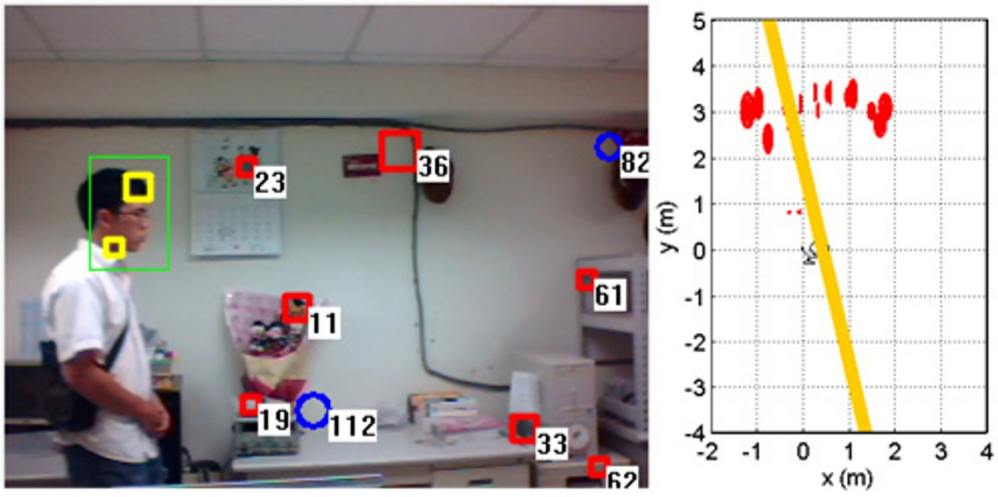

Figure 32 365th frame: moving objects are detected.

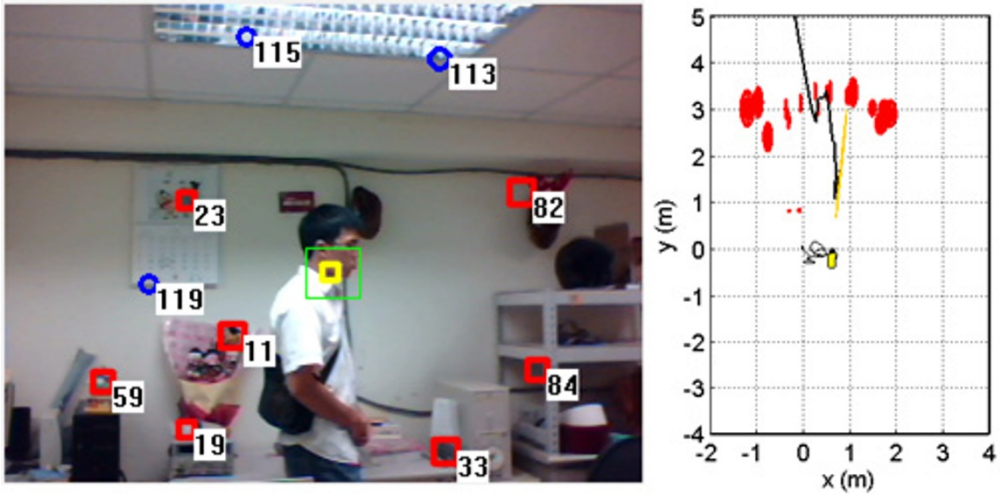

Figure 33 386th frame: the detected moving objects are tracked.

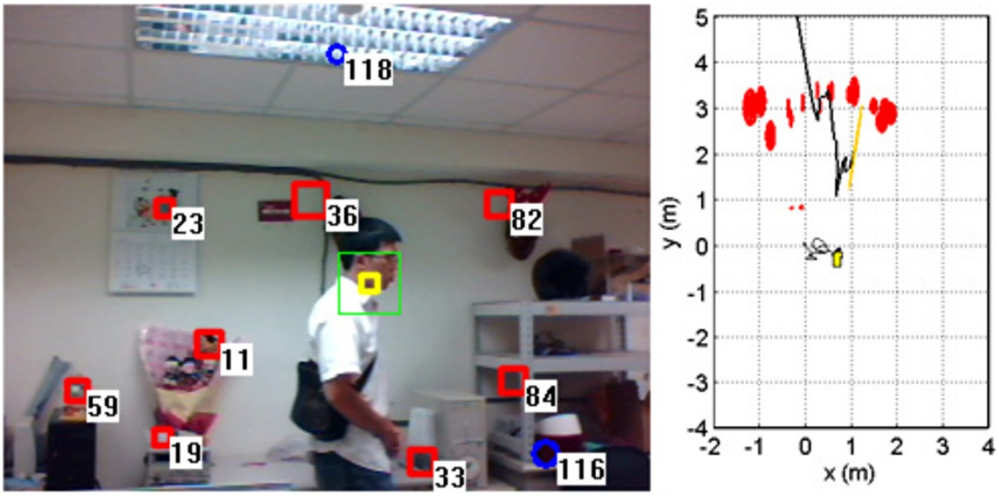

Figure 34 392nd frame: the system performs SLAM and MOT.

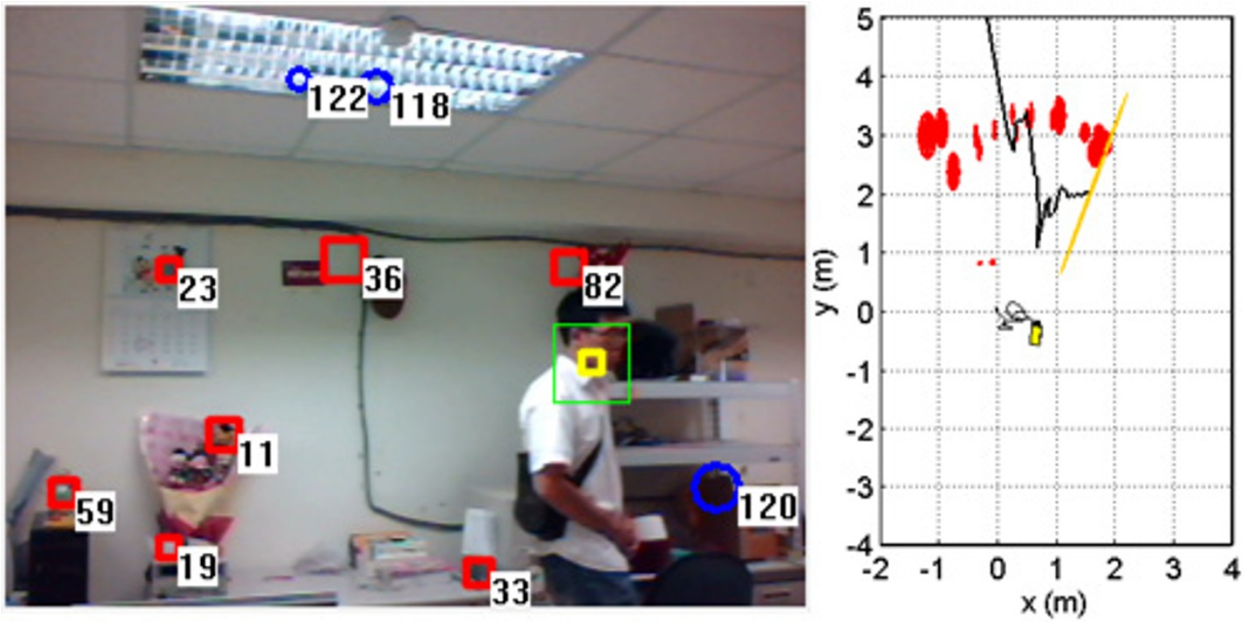

Figure 35 410th frame: MOT.

utilized to provide a robust detection of image features and a stable description of the features. Three experimental works have been carried out on a monocular vision system including SLAM in a static environment, SLAM with MOT, and people detection and tracking. The results showed that the monocular SLAM system with the proposed algorithm has the capability to support robot systems simultaneously navigating and tracking moving objects in dynamic environments.

Acknowledgements
This article was partially supported by the National Science Council in Taiwan under grant no. NSC100-2221-E-032-008 to Y.T. Wang.

Competing interests
The authors declare that they have no competing interests.

Received: 8 May 2011 Accepted: 14 February 2012
Published: 14 February 2012

References
1. CC Wang, C Thorpe, S Thrun, M Hebert, H Durrant-Whyte, Simultaneous localization, mapping and moving object tracking. Int J Robot Res. **26**(9), 889–916 (2007). doi:10.1177/0278364907081229
2. C Bibby, I Reid, Simultaneous Localisation and Mapping in Dynamic Environments (SLAMIDE) with Reversible Data Association, in *Proceedings of Robotics: Science and Systems III*, Georgia Institute of Technology, Atlanta (2007)
3. H Zhao, M Chiba, R Shibasaki, X Shao, J Cui, H Zha, SLAM in a Dynamic Large Outdoor Environment using a Laser Scanner, in *Proceedings of the IEEE International Conference on Robotics and Automation*, Pasadena, California, 1455–1462 (2008)
4. AJ Davison, ID Reid, ND Molton, O Stasse, MonoSLAM: Real-time single camera SLAM. IEEE T Pattern Anal. **29**(6), 1052–1067 (2007)
5. LM Paz, P Pinies, JD Tardos, J Neira, Large-Scale 6-DOF SLAM with Stereo-in-Hand. IEEE T Robot. **24**(5), 946–957 (2008)
6. C Harris, M Stephens, A combined corner and edge detector, in *Proceedings of the 4th Alvey Vision Conference*, University of Mancheter, 147–151 (1988)
7. N Karlsson, ED Bernardo, J Ostrowski, L Goncalves, P Pirjanian, ME Munich, The vSLAM Algorithm for Robust Localization and Mapping, in *Proceedings of the IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 24–29 (2005)
8. R Sim, P Elinas, JJ Little, A Study of the Rao-Blackwellised Particle Filter for Efficient and Accurate Vision-Based SLAM. Int J Comput Vision. **74**(3), 303–318 (2007). doi:10.1007/s11263-006-0021-0
9. DG Lowe, Distinctive image features from scale-invariant keypoints. Int J Comput Vision. **60**(2), 91–110 (2004)
10. H Bay, T Tuytelaars, L Van Gool, SURF: Speeded up robust features, in *Proceedings of the ninth European Conference on Computer Vision*, Lecture Notes in Computer Science 3951, Springer-Verlog, Berlin, German, 404–417 (2006)
11. G Shakhnarovich, T Darrell, P Indyk, *Nearest-Neighbor Methods in Learning and Vision* (MIT Press, Cambridge, MA, 2005)
12. R Smith, M Self, P Cheeseman, Estimating Uncertain Spatial Relationships in Robotics, in *Autonomous Robot Vehicles*, ed. by Cox IJ, Wilfong GT, Springer-Verlag, New York, 167–193 (1990)
13. AP Blom, Y Bar-Shalom, The interacting multiple-model algorithm for systems with Markovian switching coefficients. IEEE T Automat Control. **33**, 780–783 (1988). doi:10.1109/9.1299
14. S Hutchinson, GD Hager, PI Corke, A tutorial on visual servo control. IEEE T Robot Automat. **12**(5), 651–670 (1996). doi:10.1109/70.538972
15. L Sciavicco, B Siciliano, *Modelling and Control of Robot Manipulators* (McGraw-Hill, New York, 1996)
16. YT Wang, MC Lin, RC Ju, Visual SLAM and Moving Object Detection for a Small-size Humanoid Robot. Int J Adv Robot Syst. **7**(2), 133–138 (2010)
17. J Civera, AJ Davison, JMM Montiel, Inverse Depth Parametrization for Monocular SLAM. IEEE Trans Robot. **24**(5), 932–945 (2008)
18. T Lindeberg, Feature detection with automatic scale selection. Int J Comput Vision. **30**(2), 79–116 (1998). doi:10.1023/A:1008045108935
19. YT Wang, DY Hung, CH Sun, Improving Data Association in Robot SLAM with Monocular Vision. J Inf Sci Eng. **27**(6), 1823–1837 (2011)
20. HC Longuet-Higgins, A computer algorithm for reconstructing a scene from two projections. Nature. **293**, 133–135 (1981). doi:10.1038/293133a0
21. RI Hartley, In Defense of the Eight-Point Algorithm. IEEE T Pattern Anal. **19**(6), 580–593 (1997). doi:10.1109/34.601246

22. QT Luong, OD Faugeras, The Fundamental Matrix: Theory, Algorithms, and Stability Analysis. Int J Comput Vision. **17**(1), 43–75 (1996). doi:10.1007/BF00127818
23. JY Bouguet, Camera Calibration Toolbox for Matlab, http://www.vision.caltech.edu/bouguetj/calib_doc/ (2011)

doi:10.1186/1687-6180-2012-29
Cite this article as: Wang *et al.*: Detection of moving objects in image plane for robot navigation using monocular vision. *EURASIP Journal on Advances in Signal Processing* 2012 **2012**:29.