2005 IEEE/RSJ International Conference on Intelligent Robots and Systems

Vision-based Simultaneous Localization and Mapping with Two Cameras

Gab-Hoe Kim, Jong-Sung Kim and Ki-Sang Hong
Department of Electronic and Electrical Engineering
Pohang University of Science and Technology
San 31, Namgu, Hyojadong, Pohang, South Korea
{*ccaccafe, kimjs, hongks*}*@postech.ac.kr*

***Abstract*— In this paper, we propose a novel method for the simultaneous localization and mapping(SLAM) problem with two cameras. A single camera based approach suffers from a lack of information for feature initialization and the instability of covariance of the 3D camera location and feature position. To solve this problem, we use two cameras which move independently, unlike the stereo camera. We derive new formulations for the extended Kalman filter and map management of two cameras. We also present a method for the new features initialization and feature matching with two cameras. In our method, the covariance of camera and feature location converges more rapidly. This characteristic enables a reduction of the computational complexity by fixing the feature position whose covariance converges. Experimental results prove that our approach estimates the 3D camera location and feature position more accurately and the covariance of camera and feature location converges more rapidly when compared with the single camera case.**

***Index Terms*— SLAM, vision-based, Two cameras.**

I. INTRODUCTION

Recently, the importance of intelligent robots is increasing. One of the key components of an intelligent robot is its ability to understand its environment and recognize its position. Simultaneous localization and mapping (SLAM) deals with these problems. One powerful method to solve the SLAM problem is the extended Kalman filter based approach[1]. In the robot community, most research use the information of odometry, laser-range-finder, sonar sensor and so on. On the other hand, in the vision community, methods to use the information of camera sequences have been researched. A. Davison in [2] proposed a vision-based SLAM approach, which used active stereo head and odometry sensing for estimating the location of robot in planar region. And in [3], he has treated the localization and mapping problem using data from a single passive camera. However, the extended Kalman filter is very sensitive to outliers and increases the computational complexity in square form of the number of features. To solve these problems, T. Nir and A.M.Bruckstein[4] applied the particle filter for estimating camera pose. S.Thrun and Y.Liu[5] used the Sparse Extended Information Filters(SEIF). H.Hajjdiab et al[6] used homography. If the measurement is more accurate and the computation time is appropriately reduced, the extended Kalman filter is still powerful.

A single camera based approach lacks the information to estimate the 3D depth of feature. Thus, covariance of feature and camera is unstable. To solve the limitation of the extended Kalman filter and the lack of information using a single camera, we propose using two cameras. We can avoid improper measurement and reduce the computational complexity and get more information. We assume that two cameras are mounted on two independently moving robots, respectively, and apply the vision-based SLAM to two cameras by extending a single camera case. Our system is different from the stereo camera system, mounted on single robot, and can be extended to a vision-based multi-robot SLAM system. We derive new formulations for the extended Kalman filter and map management of the two cameras. We also present a method for the new features initialization and feature matching with two cameras. When two cameras are viewing the same scene, our method can more rapidly estimate the initial depth using the projected 1D particles, more accurately search the matching point because these particles restrict the search region in an image frame. In our method, the covariance of camera and feature location converges more rapidly. This characteristic enable the reduction of the computational complexity by fixing the feature position whose covariance converges. Experimental results prove that our approach estimates the 3D camera location and feature position more accurately and the covariance of camera and feature location converges more rapidly compared with a single camera case.

This paper is organized as follows. In section II, the vision-based SLAM algorithm with two cameras is derived. Section III explains the 2D feature matching, new feature initialization and feature management. Section IV shows

experimental results of the proposed method and compares the performance of our case to that of a single camera one. Our conclusions are given in section V.

II. SLAM WITH TWO CAMERAS

We use two cameras moving independently. This makes the movement of two cameras more flexibly and the mapping of the environment more rapidly .

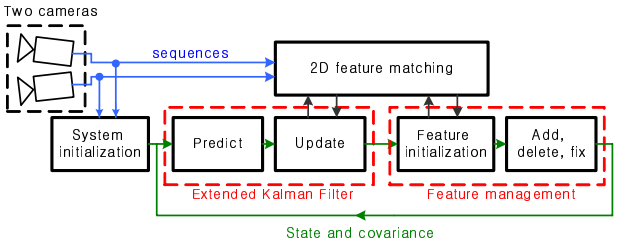


Fig. 1. The overall flow of the system.

Fig. 1 shows the overall flow of the proposed system. The system is initialized using the first frame of input sequence. In each frame, the extended Kalman filter estimates the state and covariance of each camera and features. We add the new feature and delete the inaccurate feature in the feature management part.

A. Motion model

The overall state vector $\mathbf{x}$ is composed of the camera state vector $\mathbf{c}_1$, $\mathbf{c}_2$ and the feature state vector $\mathbf{y}_i$, as

$$\mathbf{x} = \begin{pmatrix} \mathbf{c}_1^T & \mathbf{c}_2^T & \mathbf{y}_1^T & \mathbf{y}_2^T & \cdots \end{pmatrix}^T \tag{1}$$

where the camera state vector $\mathbf{c}_j$ is

$$\mathbf{c}_j = \begin{pmatrix} \mathbf{r}_j^T & \mathbf{q}_j^T & \mathbf{v}_j^T & \mathbf{w}_j^T \end{pmatrix}^T \qquad j = 1, 2. \tag{2}$$

$\mathbf{r}$ is the 3D position vector of the camera location in the world coordinate. $\mathbf{q}$ is the quaternion vector of the camera. $\mathbf{v}$ is the linear velocity and $\mathbf{w}$ the angular velocity. The feature state vector $\mathbf{y}_i$ represent the ith feature position. The covariance matrix $\mathbf{P}$ is represented as

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_{\mathbf{c}_1\mathbf{c}_1} & \mathbf{P}_{\mathbf{c}_1\mathbf{c}_2} & \mathbf{P}_{\mathbf{c}_1\mathbf{y}_1} & \mathbf{P}_{\mathbf{c}_1\mathbf{y}_2} & \cdots \\ \mathbf{P}_{\mathbf{c}_2\mathbf{c}_1} & \mathbf{P}_{\mathbf{c}_2\mathbf{c}_2} & \mathbf{P}_{\mathbf{c}_2\mathbf{y}_1} & \mathbf{P}_{\mathbf{c}_2\mathbf{y}_2} & \cdots \\ \mathbf{P}_{\mathbf{y}_1\mathbf{c}_1} & \mathbf{P}_{\mathbf{y}_1\mathbf{c}_2} & \mathbf{P}_{\mathbf{y}_1\mathbf{y}_1} & \mathbf{P}_{\mathbf{y}_1\mathbf{y}_2} & \cdots \\ \mathbf{P}_{\mathbf{y}_2\mathbf{c}_1} & \mathbf{P}_{\mathbf{y}_2\mathbf{c}_2} & \mathbf{P}_{\mathbf{y}_2\mathbf{y}_1} & \mathbf{P}_{\mathbf{y}_2\mathbf{y}_2} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}. \tag{3}$$

To predict the camera state vector $\mathbf{c}_j$, we use the function $\mathbf{f}_v(\cdot)$ as in [3]. Then, the overall state and covariance are predicted as

$$\hat{\mathbf{x}} = \begin{pmatrix} \mathbf{f}_1(\mathbf{c}_1)^T & \mathbf{f}_2(\mathbf{c}_2)^T & \mathbf{y}_1^T & \mathbf{y}_2^T & \cdots \end{pmatrix}^T, \tag{4}$$

$$\begin{aligned} \hat{\mathbf{P}}_{\mathbf{c}_1\mathbf{c}_1} &= \frac{\partial \mathbf{f}_1}{\partial \mathbf{c}_1}\mathbf{P}_{\mathbf{c}_1\mathbf{c}_1}\frac{\partial \mathbf{f}_1}{\partial \mathbf{c}_1}^T + \mathbf{Q}_1 & \hat{\mathbf{P}}_{\mathbf{c}_1\mathbf{y}_i} &= \frac{\partial \mathbf{f}_1}{\partial \mathbf{c}_1}\mathbf{P}_{\mathbf{c}_1\mathbf{y}_i} \\ \hat{\mathbf{P}}_{\mathbf{c}_2\mathbf{c}_2} &= \frac{\partial \mathbf{f}_2}{\partial \mathbf{c}_2}\mathbf{P}_{\mathbf{c}_2\mathbf{c}_2}\frac{\partial \mathbf{f}_2}{\partial \mathbf{c}_2}^T + \mathbf{Q}_2 & \hat{\mathbf{P}}_{\mathbf{c}_2\mathbf{y}_i} &= \frac{\partial \mathbf{f}_2}{\partial \mathbf{c}_2}\mathbf{P}_{\mathbf{c}_2\mathbf{y}_i} \\ \hat{\mathbf{P}}_{\mathbf{c}_1\mathbf{c}_2} &= \frac{\partial \mathbf{f}_1}{\partial \mathbf{c}_1}\mathbf{P}_{\mathbf{c}_1\mathbf{c}_2}\frac{\partial \mathbf{f}_2}{\partial \mathbf{c}_2}^T & \hat{\mathbf{P}}_{\mathbf{y}_i\mathbf{y}_j} &= \mathbf{P}_{\mathbf{y}_i\mathbf{y}_j}, \end{aligned} \tag{5}$$

where the process noise covariance $\mathbf{Q}$ is

$$\mathbf{Q}_j = \frac{\partial \mathbf{f}_j}{\partial \mathbf{n}_j}\mathbf{P}_{\mathbf{n}_j}\frac{\partial \mathbf{f}_j}{\partial \mathbf{n}_j}^T \qquad j = 1, 2. \tag{6}$$

$\mathbf{P}_n$ is the covariance of noise vector $\mathbf{n}$.

B. Measurement model

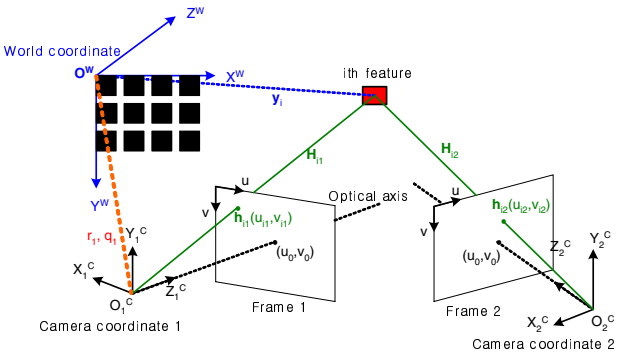


Fig. 2. The projection of ith feature onto the image frame.

In the calibrated camera, projected points of feature position $\mathbf{y}_i$ in image planes consist of $\mathbf{h}_i$(see Fig. 2), as

$$\mathbf{h}_i = \begin{pmatrix} \mathbf{h}_{i1} \\ \mathbf{h}_{i2} \end{pmatrix} = \begin{pmatrix} u_{i1} \\ v_{i1} \\ u_{i2} \\ v_{i2} \end{pmatrix} = \begin{pmatrix} u_0 - fk_u\frac{H_{i1x}}{H_{i1z}} \\ v_0 - fk_v\frac{H_{i1y}}{H_{i1z}} \\ u_0 - fk_u\frac{H_{i2x}}{H_{i2z}} \\ v_0 - fk_v\frac{H_{i2y}}{H_{i2z}} \end{pmatrix}, \tag{7}$$

$$\mathbf{H}_{i1} = \mathbf{R}_1^T(\mathbf{y}_i - \mathbf{r}_1) \qquad \mathbf{H}_{i2} = \mathbf{R}_2^T(\mathbf{y}_i - \mathbf{r}_2). \tag{8}$$

$\mathbf{y}_i$ is transformed into $\mathbf{H}_i$ in camera coordinate. $\mathbf{h}_{i1}$ and $\mathbf{h}_{i2}$ are projected ith feature $\mathbf{y}_i$ on first and second cameras. f is the focal length of camera. ku and kv represent the aspect ratio. u_0 and v_0 represent the principal point. The rotation matrix $\mathbf{R}$ is computed from quaternion $\mathbf{q}$.

C. Estimation of state and covariance

The innovation vector $\boldsymbol{\nu}_{i1}$ and $\boldsymbol{\nu}_{i2}$ are the difference between the measurement point $\mathbf{z}_i$ and prediction point $\mathbf{h}_i$ as

$$\boldsymbol{\nu}_{i1} = \mathbf{z}_{i1} - \mathbf{h}_{i1} \quad \boldsymbol{\nu}_{i2} = \mathbf{z}_{i2} - \mathbf{h}_{i2}. \tag{9}$$

Innovation covariance matrix $\mathbf{S}_i$ is

$$\mathbf{S}_i = \begin{pmatrix} \mathbf{S}_{ia} & \mathbf{S}_{ib} \\ \mathbf{S}_{ic} & \mathbf{S}_{id} \end{pmatrix} + \mathbf{R}_m \tag{10}$$

where $\mathbf{R}_m$ is the noise covariance of measurements, and $\mathbf{S}_{ia}$,$\mathbf{S}_{ib}$,$\mathbf{S}_{ic}$ and $\mathbf{S}_{id}$ of 2x2 matrix are represented as

$$\begin{aligned}
\mathbf{S}_{ia} &= \frac{\partial \mathbf{h}_{i1}}{\partial \mathbf{c}_1}\mathbf{P}_{\mathbf{c}_1\mathbf{c}_1}\frac{\partial \mathbf{h}_{i1}}{\partial \mathbf{c}_1}^T + \frac{\partial \mathbf{h}_{i1}}{\partial \mathbf{c}_1}\mathbf{P}_{\mathbf{c}_1\mathbf{y}_i}\frac{\partial \mathbf{h}_{i1}}{\partial \mathbf{y}_i}^T \\
&+\frac{\partial \mathbf{h}_{i1}}{\partial \mathbf{y}_i}\mathbf{P}_{\mathbf{y}_i\mathbf{c}_1}\frac{\partial \mathbf{h}_{i1}}{\partial \mathbf{c}_1}^T + \frac{\partial \mathbf{h}_{i1}}{\partial \mathbf{y}_i}\mathbf{P}_{\mathbf{y}_i\mathbf{y}_i}\frac{\partial \mathbf{h}_{i1}}{\partial \mathbf{y}_i}^T \\
\mathbf{S}_{ib} &= \frac{\partial \mathbf{h}_{i1}}{\partial \mathbf{c}_1}\mathbf{P}_{\mathbf{c}_1\mathbf{c}_2}\frac{\partial \mathbf{h}_{i2}}{\partial \mathbf{c}_2}^T + \frac{\partial \mathbf{h}_{i1}}{\partial \mathbf{c}_1}\mathbf{P}_{\mathbf{c}_1\mathbf{y}_i}\frac{\partial \mathbf{h}_{i2}}{\partial \mathbf{y}_i}^T \\
&+\frac{\partial \mathbf{h}_{i1}}{\partial \mathbf{y}_i}\mathbf{P}_{\mathbf{y}_i\mathbf{c}_2}\frac{\partial \mathbf{h}_{i2}}{\partial \mathbf{c}_2}^T + \frac{\partial \mathbf{h}_{i1}}{\partial \mathbf{y}_i}\mathbf{P}_{\mathbf{y}_i\mathbf{y}_i}\frac{\partial \mathbf{h}_{i2}}{\partial \mathbf{y}_i}^T = \mathbf{S}_{ic}^T \\
\mathbf{S}_{id} &= \frac{\partial \mathbf{h}_{i2}}{\partial \mathbf{c}_2}\mathbf{P}_{\mathbf{c}_2\mathbf{c}_2}\frac{\partial \mathbf{h}_{i2}}{\partial \mathbf{c}_2}^T + \frac{\partial \mathbf{h}_{i2}}{\partial \mathbf{c}_2}\mathbf{P}_{\mathbf{c}_2\mathbf{y}_i}\frac{\partial \mathbf{h}_{i2}}{\partial \mathbf{y}_i}^T \\
&+\frac{\partial \mathbf{h}_{i2}}{\partial \mathbf{y}_i}\mathbf{P}_{\mathbf{y}_i\mathbf{c}_2}\frac{\partial \mathbf{h}_{i2}}{\partial \mathbf{c}_2}^T + \frac{\partial \mathbf{h}_{i2}}{\partial \mathbf{y}_i}\mathbf{P}_{\mathbf{y}_i\mathbf{y}_i}\frac{\partial \mathbf{h}_{i2}}{\partial \mathbf{y}_i}^T.
\end{aligned} \tag{11}$$

The inverse matrix of $\mathbf{S}_i$ is defined as

$$\mathbf{S}_i^{-1} = \begin{pmatrix} \mathbf{S}_{11}^{-1} & \mathbf{S}_{12}^{-1} \\ \mathbf{S}_{21}^{-1} & \mathbf{S}_{22}^{-1} \end{pmatrix}, \tag{12}$$

then state and covariance are updated as

$$\begin{aligned}
\boldsymbol{\alpha} = \boldsymbol{\alpha} &+ \left\{ \left(\mathbf{P}_{\alpha\mathbf{c}_1}\frac{\partial \mathbf{h}_{i1}}{\partial \mathbf{c}_1}^T + \mathbf{P}_{\alpha\mathbf{y}_i}\frac{\partial \mathbf{h}_{i1}}{\partial \mathbf{y}_i}^T \right)\mathbf{S}_{11}^{-1} \right. \\
&\left. + \left(\mathbf{P}_{\alpha\mathbf{c}_2}\frac{\partial \mathbf{h}_{i2}}{\partial \mathbf{c}_2}^T + \mathbf{P}_{\alpha\mathbf{y}_i}\frac{\partial \mathbf{h}_{i2}}{\partial \mathbf{y}_i}^T \right)\mathbf{S}_{21}^{-1} \right\}\boldsymbol{\nu}_{i1} \\
&+ \left\{ \left(\mathbf{P}_{\alpha\mathbf{c}_1}\frac{\partial \mathbf{h}_{i1}}{\partial \mathbf{c}_1}^T + \mathbf{P}_{\alpha\mathbf{y}_i}\frac{\partial \mathbf{h}_{i1}}{\partial \mathbf{y}_i}^T \right)\mathbf{S}_{12}^{-1} \right. \\
&\left. + \left(\mathbf{P}_{\alpha\mathbf{c}_2}\frac{\partial \mathbf{h}_{i2}}{\partial \mathbf{c}_2}^T + \mathbf{P}_{\alpha\mathbf{y}_i}\frac{\partial \mathbf{h}_{i2}}{\partial \mathbf{y}_i}^T \right)\mathbf{S}_{22}^{-1} \right\}\boldsymbol{\nu}_{i2},
\end{aligned} \tag{13}$$

$$\begin{aligned}
\mathbf{P}_{\alpha\beta} = \mathbf{P}_{\alpha\beta} &- [\, \mathbf{P}_{\alpha\mathbf{c}_1}\{\, \mathbf{G}_{11a}\mathbf{P}_{\mathbf{c}_1\beta} + (\mathbf{G}_{11c}+\mathbf{G}_{12c})\mathbf{P}_{\mathbf{y}_i\beta} \\
&+\mathbf{G}_{12a}\mathbf{P}_{\mathbf{c}_2\beta}\,\} + \mathbf{P}_{\alpha\mathbf{y}_i}\{\, (\mathbf{G}_{11b}+\mathbf{G}_{21b})\mathbf{P}_{\mathbf{c}_1\beta} \\
&+(\mathbf{G}_{11d}+\mathbf{G}_{12d}+\mathbf{G}_{21d}+\mathbf{G}_{22d})\mathbf{P}_{\mathbf{y}_i\beta} \\
&+(\mathbf{G}_{12b}+\mathbf{G}_{22b})\mathbf{P}_{\mathbf{c}_2\beta}\,\} + \mathbf{P}_{\alpha\mathbf{c}_2}\{\, \mathbf{G}_{21a}\mathbf{P}_{\mathbf{c}_1\beta} \\
&+(\mathbf{G}_{21c}+\mathbf{G}_{22c})\mathbf{P}_{\mathbf{y}_i\beta} + \mathbf{G}_{22a}\mathbf{P}_{\mathbf{c}_2\beta}\,\}\,],
\end{aligned} \tag{14}$$

where $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ represent camera state or each feature state vector and matrix $\mathbf{G}$ is represented as

$$\begin{aligned}
\mathbf{G}_{jka} &= \frac{\partial \mathbf{h}_{ij}}{\partial \mathbf{c}_j}^T\mathbf{S}_{jk}^{-1}\frac{\partial \mathbf{h}_{ik}}{\partial \mathbf{c}_k} \quad \mathbf{G}_{jkb} = \frac{\partial \mathbf{h}_{ij}}{\partial \mathbf{y}_i}^T\mathbf{S}_{jk}^{-1}\frac{\partial \mathbf{h}_{ik}}{\partial \mathbf{c}_k} \\
\mathbf{G}_{jkc} &= \frac{\partial \mathbf{h}_{ij}}{\partial \mathbf{c}_j}^T\mathbf{S}_{jk}^{-1}\frac{\partial \mathbf{h}_{ik}}{\partial \mathbf{y}_i} \quad \mathbf{G}_{jkd} = \frac{\partial \mathbf{h}_{ij}}{\partial \mathbf{y}_i}^T\mathbf{S}_{jk}^{-1}\frac{\partial \mathbf{h}_{ik}}{\partial \mathbf{y}_i}.
\end{aligned} \tag{15}$$

When compared with a single camera SLAM, two cameras inject a extra term which correlates with the covariance of the first and second camera into (13) and (14). A more accurate position of the camera and a more rapid convergence of covariance are given by this extra term.

III. Feature Management

The extended Kalman filter uses known features which are manually selected. As cameras move, these features could disappear out of the camera's view, so that we need to add a new feature and delete an inaccurate feature.

A. Initialization of new feature

As cameras move, a new feature is detected using Harris-Stephens corner detector[7] in each frame. However, we can't know the 3D position of a detected feature because the feature depth is unknown. Hence, we estimate the initial position of the feature using some frames and particles. 100 particles in the range 0.5m to 5.0m are created on the 3D line. In the next frame, the matching point is found surrounding projected particles using the NSSD(Normalized Sum of Squared Difference) . Then the interval of particles is reduced as shown in Fig. 3.

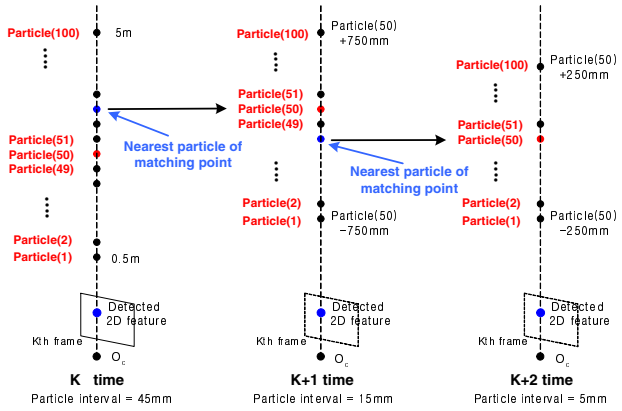


Fig. 3. Generation of 1D particles.

If particles are projected on just one camera, the initial 3D position of a new feature is determined by the weighted sum of the inverse distance between the projected particles and the matching point after the interval is reduced to 5mm. The weighted sum is calculated in 3 consecutive frames. If the particle is projected on two cameras, we find the matching point in only a small search region which is defined by the projected particles. The initial 3D position of a new feature is accurately estimated by the triangulation[9] using each matching point. This approach makes the extended Kalman filter more stable.

B. 2D feature matching

We store 15x15 patch whenever the initial feature is detected in each frame and find the matching point in the limited search region. The search region is affected by the information of uncertainty of the projected point which is represented by the covariance matrix $\mathbf{S}_{ia}$ in the first camera and $\mathbf{S}_{id}$ in the second camera as in (11). Both $\mathbf{S}_{ia}$ and $\mathbf{S}_{id}$ are the covariance of 2D Gaussian distribution. Choosing a number of standard deviations defines an elliptical search

region within which the feature should lie with high probability. Thus, we can get the matching point in the elliptical search region, without another feature tracking algorithm. If the shape of the feature patch is very similar to other patches in the search region, the NSSD value may not be reliable. To prevent this case, we discard the feature if the number of similar NSSD values is over the threshold. Viewing the same feature in two cameras, we also discard the feature if the feature matching point is detected in the search region of just one camera.

C. Feature addition and deletion

If the nth new feature is added, we add the new state $\mathbf{y}_n$ to the overall state as

$$\mathbf{x} = \begin{pmatrix} \mathbf{c}_1^T & \mathbf{c}_2^T & \mathbf{y}_1^T & \mathbf{y}_2^T & \cdots & \mathbf{y}_n^T \end{pmatrix}^T. \tag{16}$$

Using the first order Taylor expansion and new state, the added covariance is represented as

$$\begin{aligned} \mathbf{P}_{\mathbf{y}_n\boldsymbol{\alpha}} &= \frac{\partial \mathbf{y}_n}{\partial \mathbf{c}_1}\mathbf{P}_{\mathbf{c}_1\boldsymbol{\alpha}} + \frac{\partial \mathbf{y}_n}{\partial \mathbf{c}_2}\mathbf{P}_{\mathbf{c}_2\boldsymbol{\alpha}} \\ &\quad (\boldsymbol{\alpha} = \mathbf{c}_1, \mathbf{c}_2, \mathbf{y}_1, \cdots, \mathbf{y}_{n-1}), \end{aligned} \tag{17}$$

$$\begin{aligned} \mathbf{P}_{\mathbf{y}_n\mathbf{y}_n} &= \frac{\partial \mathbf{y}_n}{\partial \mathbf{c}_1}\mathbf{P}_{\mathbf{c}_1\mathbf{c}_1}\frac{\partial \mathbf{y}_n}{\partial \mathbf{c}_1}^T + \frac{\partial \mathbf{y}_n}{\partial \mathbf{c}_2}\mathbf{P}_{\mathbf{c}_2\mathbf{c}_1}\frac{\partial \mathbf{y}_n}{\partial \mathbf{c}_1}^T \\ &\quad + \frac{\partial \mathbf{y}_n}{\partial \mathbf{c}_1}\mathbf{P}_{\mathbf{c}_1\mathbf{c}_2}\frac{\partial \mathbf{y}_n}{\partial \mathbf{c}_2}^T + \frac{\partial \mathbf{y}_n}{\partial \mathbf{c}_2}\mathbf{P}_{\mathbf{c}_2\mathbf{c}_2}\frac{\partial \mathbf{y}_n}{\partial \mathbf{c}_2}^T \\ &\quad + \frac{\partial \mathbf{y}_n}{\partial \mathbf{H}_{n1}}\mathbf{R}_H\frac{\partial \mathbf{y}_n}{\partial \mathbf{H}_{n1}}^T + \frac{\partial \mathbf{y}_n}{\partial \mathbf{H}_{n2}}\mathbf{R}_H\frac{\partial \mathbf{y}_n}{\partial \mathbf{H}_{n2}}^T, \end{aligned} \tag{18}$$

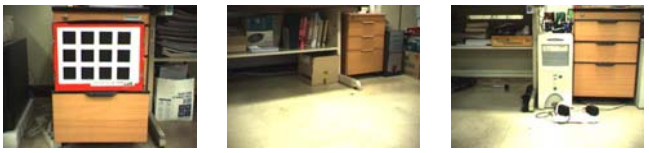
where $\mathbf{R}_H$ is the measurement noise $\mathbf{R}_m$ transformed into 3D space.

A feature is deleted from the state if more than a certain threshold are failures. In the covariance matrix, we also remove the rows and columns which are related to the deleted features. In our method the covariance of feature location converges more rapidly. The 3D position of this feature is nearly unchanged. To reduce the computational complexity, we fix the feature position whose covariance is converged. The covariance of fixed feature is set to zero.

IV. Experimental results

In our experimental implementation, we used a Point Gray $Dragonfly^{TM}$ camera with a resolution of 640x480 pixels. The Z.Zhang's algorithm[8] is used to calibrate the camera. We got two sequences with only one camera(given Figs. 4 and 5). In order to get the initial camera position, we used the pattern of which 3D position is known in world coordinate as in Fig. 4(a).

After the initial features were manually selected, initial state is calculated by using the 8 known features extracted from the given pattern. Because we assume that two cameras are mounted on two independently moving robots, the motion of each camera should be different. Camera motion was controlled by hand in this experiment. The first camera moves in an arc and the second camera moves along the diagonal line in 0.9x0.9m square. And then, both cameras return to



(a) 1st frame (b) 400th frame (c) 600th frame

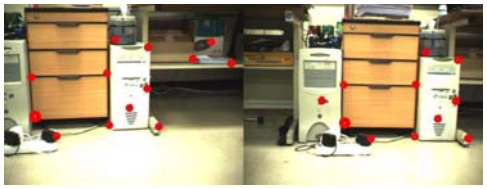
Fig. 4. The snapshots of a sequence from the first camera.

(a) 1st frame (b) 400th frame (c) 600th frame

Fig. 5. The snapshots of a sequence from the second camera.

(a) 1st frame

(b) 200th frame

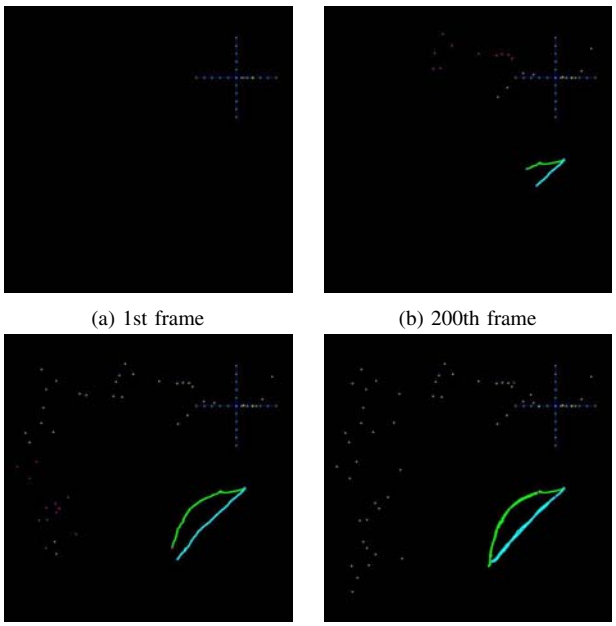


(c) 500th frame

(d) 1000th frame

Fig. 6. The elliptical search regions(red) to measure the features between first camera(left) and second one(right).

the initial position. In the proposed method, the results of estimating the state and covariance of camera and features are represented in Figs. 6 and 7. The red elliptical region in Fig. 6 represents search region. Fig. 7 shows the camera motion and feature position viewing on the top.



(a) 1st frame (b) 200th frame

(c) 500th frame (d) 1000th frame

Fig. 7. 2D maps of the camera motion and feature positions on XZ-plane. The blue lines are X and Z axis. The green line is the motion of the first camera and the cyan line is the motion of the second camera. Spots are features.

A. The performance comparison of the proposed method and a single camera one

Fig. 8(a) shows the covariance of a camera position using a single camera SLAM algorithm[3] and Fig. 8(b) shows the one of the first camera(same motion in the single camera SLAM) using the proposed method. In a single camera case, the covariance of the camera remains unstable, while the proposed method shows more stable covariance of the camera. Similarly, Fig. 10 shows that the covariance of quaternion is more stable compared with a single camera case. Not only the covariance of the camera state, the covariance of the feature state also shows that the covariance converges more rapidly. Fig. 9 represents the covariance of feature position during 50 frames at first 10 features except 8 known features. The result map of the camera motion and feature position is represented in Fig. 11. The red line is the camera motion of a single camera and the blue line is the first one of two cameras. The camera motion with two cameras is more accurate than a single camera. The feature position(blue point) with two cameras is also more reliable than a single camera(red point). In a single camera SLAM algorithm, the extended Kalman filter uses full covariance matrix. As the number of features increases, the computational complexity increase. In our method the covariance of camera and feature location converges more rapidly. This characteristic reduces the computational complexity by fixing the feature position whose covariance converges. The covariance of fixed feature is set to zero. The process time with two cameras is not over 0.2 second in each frame(see Fig. 12).

V. Conclusion

We described a novel method for the simultaneous localization and mapping(SLAM) problem with two cameras which move independently. In this paper, there are several contributions: Firstly, we derived new formulations for the extended Kalman filter and map management of two cameras. Secondly, we also presented a method for the new features initialization and the feature matching with two cameras. Finally, computational complexity was reduced by fixing the feature position whose covariance was converged.

In the future, we will focus on research using multi-cameras and tracking them over a greater distance. We also plan to design an effective feature management scheme, which would allow proper distribution of features and improve their availability.

References

[1] G. Dissanayake, P. Newman, et al, "A Solution to the simultaneous localization and amp building (SLAM) problem", IEEE Transactions on Robotics and Automation, 17(3):229-241, 2001.

[2] A. J. Davison and D. W. Murray, "Simultaneous localization and map-building using active vision", IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7):865-880, 2002.

[3] A. J. Davison, "Real-Time Simultaneous Localisation and Mapping with a Single Camera", Proceedings of the 9th International Conference on Computer Vision, pp:1403-1410, 2003.

[4] T. Nir and A. M. Bruckstein, "Causal Camera Motion Estimation by Condensation and Robust Statistics Distance Measures", Proceedings of the 8th European Conference on Computer Vision, 2004.

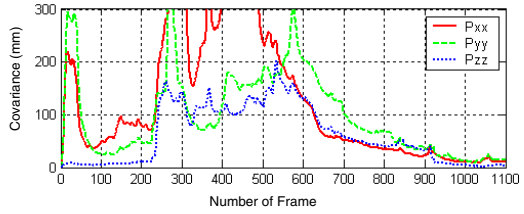
[5] S. Thrun and Y. Liu, "Multi-Robot SLAM With Sparse Extended Information Filters", Proceedings of the 11th International Symposium of Robotics Research, 2003.

[6] H. Hajjdiab and R. Laganiere, "Vision-based Multi-Robot Simultaneous Localization and Mapping", Proceedings of the First Canadian Conference on Computer and Robot Vision, pp:155-162, 2004.

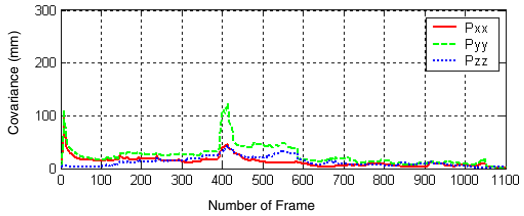
[7] C. Harris and M. J. Stephens, "A combined corner and edge detector", In Alvey Vision Conference, pp:147-152, 1988.

[8] Z. Zhang, "A flexible new technique for camera calibration", IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11):1330-1334, 2000.

[9] R. Hartley and A. Zisserman "Multiple View Geometry", Cambridge Univ. press, pp:295-298, 2000

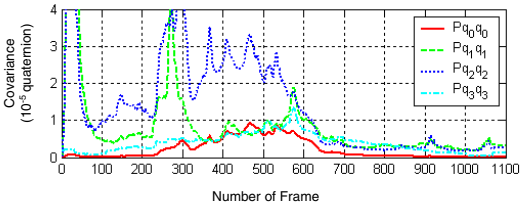


(a) A single camera

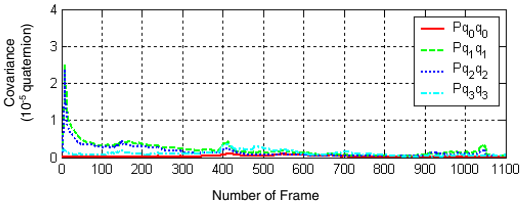


(b) Two cameras

Fig. 8. The covariance of camera position. The camera location is more stable in (b) compared with (a).

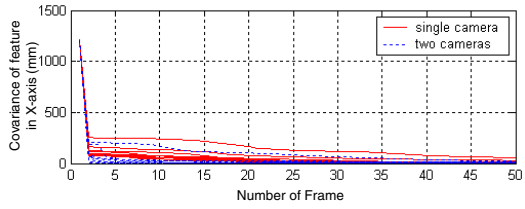


(a) A single camera

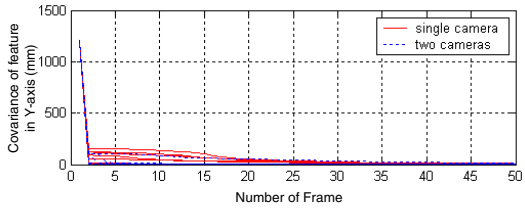


(b) Two cameras

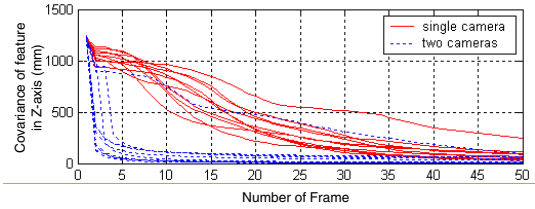
Fig. 10. The covariance of camera quaternion. The camera rotation is more stable in (b) compared with (a).



(a) Feature covariance in X-axis $\mathbf{P}_{xx}$



(b) Feature covariance in Y-axis $\mathbf{P}_{yy}$



(c) Feature covariance in Z-axis $\mathbf{P}_{zz}$

Fig. 9. The comparison of covariances from the first 10 features. The feature covariances converges more rapidly in two cameras.

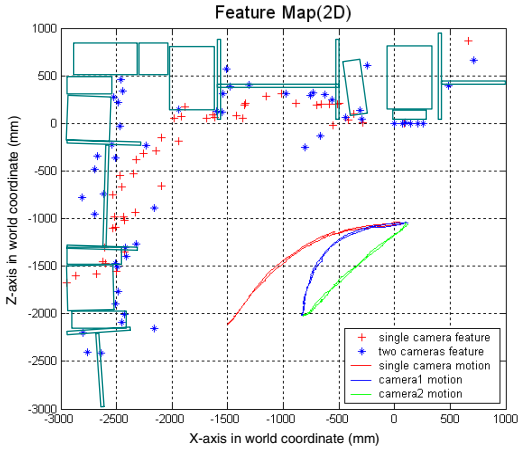


Fig. 11. The final 2D map of camera motions and features on XZ-plane. In the same camera, the camera motion is more accurate in one of the two cameras(blue) compared with a single camera(red).

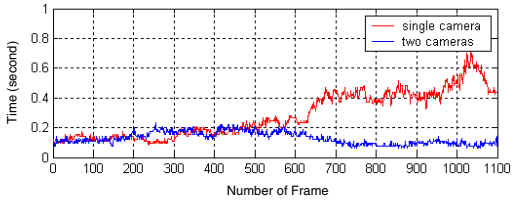


Fig. 12. The computation time. The time of the proposed method is not over 0.2 second in each frame.