SELF-LOCATION ESTIMATION OF A MOVING CAMERA
USING THE MAP OF FEATURE POINTS AND EDGES OF ENVIRONMENT

TOSHIYUKI ADACHI, KATSUYA KONDO, SYOJI KOBASHI and YUTAKA HATA,
GRADUATE SCHOOL OF ENGINEERING, UNIVERSITY OF HYOGO, JAPAN

ABSTRACT
Applications as visual navigation of mobile robot with image sensor and mixed/augmented reality have been investigated actively. Many of these techniques require the localization of human or robot. In this report, we propose a novel method for estimating the position of a camera by using edge and feature point information in time-series of images. The technique can be applied to the indoor environment and the environment spatial model including feature points is obtained by tracking background feature points with the position of camera. The prediction model image is generated based on trajectory of the moving camera from the spatial model and the camera position is estimated by correction of model image from comparison with the input image.

KEYWORDS: Camera Position, Feature Point, Feature Edge, Monocular Vision, Moving Camera, Frame Model

1. INTRODUCTION
In recent years, many researchers have studied to reconstruct 3-dimensional(3D) scene[1], and navigate moving robots with visual sensor by using indoor and outdoor images obtained from cameras[2][3]. In these techniques, 3D positional information of a camera is required. As the conventional methods to estimate camera position, there are techniques using sensor such as GPS and high speed range sensor[4]. In the methods with GPS, camera position is estimated by latitude and longitude information. However, since methods with high speed range sensor require special equipments. Furthermore, methods with GPS cannot be used in indoor environments. Therefore, above problems are solved when camera position is estimated by using only images without special sensor. Other methods estimate camera position by using the image sequence[5]. The method is to make feature points of a frame in image sequence match points of other frame, and it calculates epipola line and to estimate camera position by pseudo stereo of image sequence. This method has some problems of mismatch between feature points and accumulative error. Another method to estimate camera location by tracking natural feature points and markers whose 3D location is previously known, has been proposed[6]. This method takes fitting process of whole feature points for reducing accumulative error. Furthermore, there are methods using feature points for landmarks[7][8]. In the methods, extracted feature points, the set of landmarks corresponding to camera parameters and the image texture around landmarks are put in database. Then, camera position is estimated by matching the input image to the database.

We proposed the method to estimate the camera self-location from extracted features in time-series of images using single-eyed camera by edge information of indoor environment[9]. But this method is not able to estimate the camera self-location when the corner of the environment is not taken. In this paper, we propose the method to estimate the camera self-location continuously. In our method, we extract feature points of background and make the distribution map of feature points as the wall model from the image and estimated camera parameters. Using the map of feature points of walls, the move of camera is tracked in term when the corner is not taken.
2. PRELIMINARY

In our experimental system, we move one single-eyed handy camera in the rectangular parallelepipeds environment. There is no object covering backgrounds. The focal length of camera and the first camera position and pose are known. We have indoor model framework. The experimental environment and our coordinate system are shown in Figure 1. The pan of the camera pose is the Y-axis rotation angle, the tilt is the X-axis rotation angle and the screw is the Z-axis rotation angle in the camera coordinate. We show assumptions in our method below.
1. Change in the screw angle is small enough.
2. First frame of input images include vertical line connecting ceiling and floor of environment.
3. Lighting condition is permanent.

3. METHOD

3.1 Prediction of Camera Position

We predict the camera position and pose in the next frame with an estimated trajectory of moving camera position \( C_i \) and pose \( \Theta_i \). Assuming camera moving smoothly, the camera position and pose are predicted by Eq.(1) and (2) using previous three frames. We set pan angle of camera to \( \theta \) and tilt angle of camera to \( \phi \).

\[
(C_i - C_{i-1}) - (C_{i-1} - C_{i-2}) = (C_{i-1} - C_{i-2}) - (C_{i-2} - C_{i-3}) \tag{1}
\]

\[
(\Theta_i - \Theta_{i-1}) - (\Theta_{i-1} - \Theta_{i-2}) = (\Theta_{i-1} - \Theta_{i-2}) - (\Theta_{i-2} - \Theta_{i-3}) \tag{2}
\]

3.2 Estimation of Camera Position with Edge

3.2.1 Comparison between Framework and Edge

We transpose framework by predicted camera position and pose. We extract edge line on the input image using sobel filter and limit extracted edges as the same color area as necessary edge line. Edge lines show border of wall, ceiling and floor. Extracted edge lines are shaded off by gaussian filter. We transpose framework by using the perspective projection and predicted camera position and pose. Relation between 3D position \( M_o \) in camera coordinate and 3D position \( M_w \) in world coordinate is shown in Eq.(3).

\[
M_w = R(M_o - C_{pos}) \tag{3}
\]

\[
C_{pos} = [C_x \ C_y \ C_z]^T, \quad R = \begin{bmatrix}
\cos\theta & 0 & \sin\theta \\
0 & 1 & 0 \\
-\sin\theta & 0 & \cos\theta
\end{bmatrix}
\]

We put transposed framework on edges in input image, and compare position of edge lines. Searching areas shown in Figure 3 are set at distance of four pixels from the projected vertex of framework. We define points overlapped with transposed framework and image rectangular frame as end-points of edge line. We also define the vertex as end-point of vertical edge line connecting ceiling and floor. Next, we set the vertex as overlapping point horizontal line and image rectangular framework. Then, we repeat searching process until Eq.(4) is satisfied. When framework match with extracted edge, lines connecting end-points are decided as true edge line.
\[ \sum_{j_{frame}} |I_{frame} - I_{edge}| < \text{threshold} \cdot \text{Pixel}_{frame}, \text{threshold} = 0.8 \] (4)

where \( I_{frame} \) and \( I_{edge} \) is respectively the value of pixel of projected model and extracted edge.

### 3.2.2 Correction of Camera Position

We calculate the image coordinates of end-points of the edge line which each slope of two lines growing sideways from points. \( \alpha \) is the angle of X-axis and \( \beta \) is the angle of Y-axis shown in Figure 4.

\[
\phi = \sin^{-1} \frac{\Delta W_y \cos \theta}{\sqrt{(\Delta W_y - \Delta W_x \sin \theta)^2 + (\Delta W_x - \Delta W_y \sin \theta)^2}} - \alpha
\] (5)

\[
a = \tan^{-1} \frac{\Delta W_x - \Delta W_z \sin \theta}{\Delta W_x - \Delta W_y \sin \theta}
\]

where \((W_{ix}, W_{iy}, W_{iz})\), \((W_{ix}, W_{iy}, W_{iz})\) are coordinates the end-point in world coordinate.

We calculate camera angle from the vertical line using the image lines. And we calculate slope of four horizontal edge lines. \( S_1 \) and \( S_2 \) are slope of horizontal line of upper corner. \( S_3 \) and \( S_4 \) are slope of horizontal line of lower corner. We estimate camera position and pose from slope of horizontal edge lines by Eq.(6).

\[
C_x = \left( \frac{S_y W_{ix} - S_x W_{iy}}{S_x W_{ix} - S_y W_{ix}} \right) \left( \frac{S_y W_{iy} - S_x W_{ix}}{S_x W_{iy} - S_y W_{iy}} \right) \]

(6-a)

\[
C_y = \left( \frac{S_x W_{ix} - S_y W_{iy}}{S_x W_{ix} - S_y W_{ix}} \right) \left( \frac{S_x W_{iy} - S_y W_{ix}}{S_x W_{iy} - S_y W_{iy}} \right) \]

(6-b)

\[
C_z = \left( \frac{S_y W_{iz} - S_x W_{iz}}{S_x W_{iz} - S_y W_{iz}} \right) \left( \frac{S_y W_{iz} - S_x W_{iz}}{S_x W_{iz} - S_y W_{iz}} \right) \]

(6-c)

Then, we calculate camera pose by (7), (13).

\[
\theta = \sin^{-1} \left( \frac{\sin \beta (W_{ix} - C_x) - \sin \beta (W_{ix} - C_x)}{\sin \beta (C_x + \cos \beta - \sin \beta \cos \beta)} \right)
\]

(7)

\[
a_x = \tan^{-1} \left( \frac{S_x}{S_y} \right), a_y = \tan^{-1} \left( \frac{S_x}{S_y} \right), \beta_x = \tan^{-1} \left( \frac{S_x}{S_y} \right), \beta_z = \tan^{-1} \left( \frac{S_x}{S_y} \right)
\]

### 3.3 Renewal of Environmental Model

In this method, we use six background textures of wall, floor and ceiling as a rectangular parallelepiped environment model. Background textures are obtained by projecting feature points extracted from the image on background plains with estimated value of camera position and pose shown in Figure 5.

### 3.4 Estimation of Camera Position with Feature Point

#### 3.4.1 Comparison between Environment Model and Feature Point

Feature points extracted from the image with SUSAN operator when the corner cannot be taken. With predicted camera position and pose, environmental model is projected to the image and the image coordinate of feature points of model corresponding to camera position. Matching
extracted points to model points, the image coordinate of those are compared. From the image, points which is not exactly feature points are extracted Then, searching area is prepared on 4-pixel distance from projected model points. Model point of center of the searching area and extracted points in the area are regarded as same feature of background.

3.4.2 Correction of Camera Position

The camera position is estimated by using world coordinate and image coordinate of feature points and predicted camera position. The camera pose is estimated by using this estimated camera position. The number of feature points on the image is not estimation process is separated to estimation of camera position and camera pose, to estimate camera position and pose efficiently. The angle between the feature point and camera from image coordinate of the point is calculated. The angle $\alpha$ of X-axis and the angle $\beta$ of Y-axis between the feature point and camera are respectively calculated. The vector from the feature point to camera is calculated by using the angle between the feature point and camera and predicted camera pose.

$$\overline{PC}_{posi} = R R_R [0 \ 0 \ 1]^T$$ (8)

The feature point and the camera are on the straight line shown in Eq(9).

$$\frac{P_{ix} - C_x}{l} = \frac{P_{iy} - C_y}{m} = \frac{P_{iz} - C_z}{n}$$ (9)

An intersection point of two straight lines connecting feature point and camera is camera position. Then, camera position is estimated from Eq(10) by using two feature points $P_i, P_j (i \neq j)$.

$$C_x = \left( \frac{m_i}{l_i} P_{ix} - \frac{m_i}{l_i} P_{iz} + P_{ix} - P_{iz} \right) \left( \frac{l_i}{l_i} \frac{l_i}{l_i} \right)$$ (10-a)

$$C_y = \left( \frac{n_i}{m_i} P_{iy} - \frac{n_i}{m_i} P_{iy} + P_{iy} - P_{iy} \right) \left( \frac{n_i}{m_i} \frac{n_i}{m_i} \right)$$ (10-b)

$$C_z = \left( \frac{l_i}{n_i} P_{iz} - \frac{l_i}{n_i} P_{iz} + P_{iz} - P_{iz} \right) \left( \frac{l_i}{n_i} \frac{l_i}{n_i} \right)$$ (10-c)

where $(C_x, C_y, C_z)$ is the camera position of world coordinate, $(P_{ix}, P_{iy}, P_{iz})$ is the feature points position of world coordinate. From these processes, candidates of the estimated camera position $C_{est}(C_{estx}, C_{esty}, C_{estz})$ corresponding to combinations of two feature points. The predicted camera position is modified to the end result of the estimated camera position by using estimated candidates. The vector from predicted camera position to estimated candidates of camera position is calculated, and the vector weighted corresponding to the absolute value of that. The means of weighted vectors as modified value added to predicted camera position, and the end result of the estimated camera position is calculated by Eq.(11).

$$\hat{C}_{posi} = X + \frac{\sum X_i C_i a_i}{\sum a_i C_i} \quad a = 0.95$$ (11)

3.4.3 Correction of Camera Pose

The camera pose is estimated by using world coordinate and image coordinate of feature points and predicted camera pose. The vector from estimate position $\hat{C}_{posi}$ to the feature point is shown in Eq.(12). Eq.(8) rotation matrix is shown in Eq.(13).

$$\begin{vmatrix} l \\ m \\ n \end{vmatrix} = \begin{vmatrix}1 \\ \sqrt{(C_x - P_x)^2 + (C_y - P_y)^2 + (C_z - P_z)^2} \\ \sqrt{(C_x - P_x)^2 + (C_y - P_y)^2 + (C_z - P_z)^2} \end{vmatrix} \begin{vmatrix}C_x - P_x \\ C_y - P_y \\ C_z - P_z \end{vmatrix}$$ (12)

$$R_{RP} = \begin{vmatrix}0 \\ 1 \\ 0 \end{vmatrix} = \begin{vmatrix}1 \\ \sqrt{(C_x - P_x)^2 + (C_y - P_y)^2 + (C_z - P_z)^2} \\ \sqrt{(C_x - P_x)^2 + (C_y - P_y)^2 + (C_z - P_z)^2} \end{vmatrix} \begin{vmatrix}C_x - P_x \\ C_y - P_y \\ C_z - P_z \end{vmatrix}$$ (13)
The pan angle and the tilt angle of camera is shown in Eq.(14), by substituting the estimated camera position and the world coordinate of feature points for Eq.(12).

\[
\begin{align*}
\theta &= \sin^{-1} \left( \frac{\sin \beta (P_n - C_x) - \sin \beta (P_n - C_x)}{\sin \beta \cos \theta \cos \beta - \sin \beta \cos \alpha \cos \beta} \right) \quad (14-a) \\
\phi &= \sin^{-1} \left( \frac{\cos \theta \sin \alpha \sin \beta + (P_n - C_x)}{\sin \theta - \cos \theta \cos \alpha \cos \beta} \right) \quad (14-b)
\end{align*}
\]

From these processes, candidates of the estimated camera pose corresponding to combinations of two feature points. Similarly to the estimation of the camera position, the predicted camera pose is modified to the end result of the estimated camera position by using estimated candidates. The means of weighted vectors from predicted pose as modified value added to predicted camera pose \( \hat{C}_{pose} \), and the end result of the estimated camera pose is calculated by Eq.(11).

4. EXPERIMENTAL RESULTS

In this chapter, we show the experimental results. We move one single-eyed handy camera (Panasonic, NV-GS400K, 720×480, 30fps) in the rectangular parallelepips gymnasium. The resolution of the camera is 720 x 480 pixels.

4.1 Position of a Stationary Camera

Camera is set on a tripod and camera pose is freely changed. Figure 6, 7 shows trajectory of camera position. The camera parameters are \( C_x = 4.00 \text{m}, \; C_y = 1.60 \text{m} \) and \( C_z = 4.00 \text{m} \). From the result, at term of estimation by feature point tracking from #550 to #2800 frame the error is increase but at term of estimation by edge information the error is decrease.

4.2 Pose of a Stationary Camera

Camera is set on a truck and camera position is changed in straight line. Figure 8, 9 shows trajectory of camera position. The camera parameters are \( \theta = -12.0 \text{deg} \) and \( \phi = 0.0 \text{deg} \). From the result, camera pose is estimated stably to frame sequence.

4.3 Self-Location of a Moving Camera

Camera is moved in the hand and image is taken. Trajectories of camera position are shown in Figure 11. To check the camera trajectory is estimated well, we moved the camera on circular orbit. From results of the estimated trajectory is certainly a circular orbit. And figure 11 shows composite images of input images and virtual images generated by estimated camera parameters.

5. CONCLUSIONS

In this paper, we proposed the estimation method of camera position by using time series of images taken by the camera. In our method, we can obtain the position of the single camera by tracking of the feature point at the term when estimation by the edge information is disabled. Our method can apply not only room as shown in experiment, but also various fields such as gymnasium. From experimental results, our method can continuously estimate the self-location of a stationary and moving camera whichever the corner is taken, or not. Our future work is to estimate the position of camera with complex motion and to estimate it in complex environments.

6. REFERENCES