The Third Joint Workshop on
Machine Perception and Robotics
(MPR2007)

Environmental Mapping by Trinocular Vision for Self-Localization Using Monocular Vision

Yoko OGAWA, Nobutaka SHIMADA, Yoshiaki SHIRAI
Ritsumeikan University,
1-1-1 Noji-higashi, Kusatu, Shiga, Japan

Abstract- **This paper presents a SIFT based map building and self localization method for mobile robots. 3-D positions of landmarks are useful for the robot to localize the self position, 3-D input system occupies larger space and computational cost is too high to implement on the embedded system. Although motion stereo techniques using only a monocular vision system have been developed, the accuracy of the depth measurement is not enough to build navigation maps. We proposed the following scheme: First a mapping robot with trinocular camera builds the 3-D keypoint map of unknown environment with a high accuracy and then the working robot with a monocular camera localize the own position by matching the SIFT keypoints to the map. Experimental results of mapping and localization for a real indoor scene are shown.**

I. Introduction

Automatic environmental map building is necessary for mobile robot navigation in unknown environment. There has recently been considerable progress in developing real-world environment mapping system based on the use of visual appearances of natural or artificial landmark features [1, 2, 3, 4]. While 3-D positions of landmarks are useful for the robot to localize the self position, the input system occupies larger space and computational cost is too high to implement on the embedded system. Although motion stereo techniques using only a monocular vision system have been developed, the accuracy of the depth measurement is not enough to build navigation maps. Since most of significant landmarks in indoor scenes are not often changed, the following scheme is proposed: at first a mapping robot with trinocular vision once explores the environment and builds the map of high quality, then working robot with a monocular vision moves in the environment by matching the visual cues to landmarks in the map. This paper described the above scheme and shows experimental results of map building and navigation in a real scene.

Here egomotion is estimated only from the image feature because we use an omni directional vehicle where installation the rotary encoder is difficult.

II. Extraction and Tracking Keypoint Using SIFT

In order to initially explore the environment, we utilize a trinocular camera system of L shaped configuration (Digiclops by Point Gray Research Inc.). From the three captured images at each time frame, SIFT (Scale Invariant Feature Transform) [5, 1] keypoints are extracted as landmarks. Trinocular correspondences are made to obtain their depths. The keypoints are also tracked over the time sequence. The robot's egomotion and the 3-D positions of the features are estimated and then registered on the map.

A. Feature Extraction and Depth Measurement

The SIFT keypoints are extracted from DoG (Deference of Gaussian) dimension extreme over a threshold. Each SIFT keypoint has image coordinate $[x\ y]^T$, scale s, orientation o, contrast c and feature vector $\mathbf{f}$. The feature vector is a 4x4x8 dimension vector which represents local intensity patterns of the neighbor region.

The unique correspondences are searched for betweentwo images. A pair (right-left and right-top) of unique correspondence composes a stable keypoint. Candidatesof corresponding pairs of keypoints should satisfy the following constraints:

- the epipolar constraints
- the difference of the scales is within 1 level
- the difference of the orientation is less than 20 degrees
- the distance between the SIFT feature vectors below a threshold

Correspondences are decided based on the distance between the feature vectors among those candidates. If the shortest distance is smaller than 70% of that of the second-shortest, the correspondence is established between the shortest pair.

The stable keypoints have a pair (right-left and right-top) of

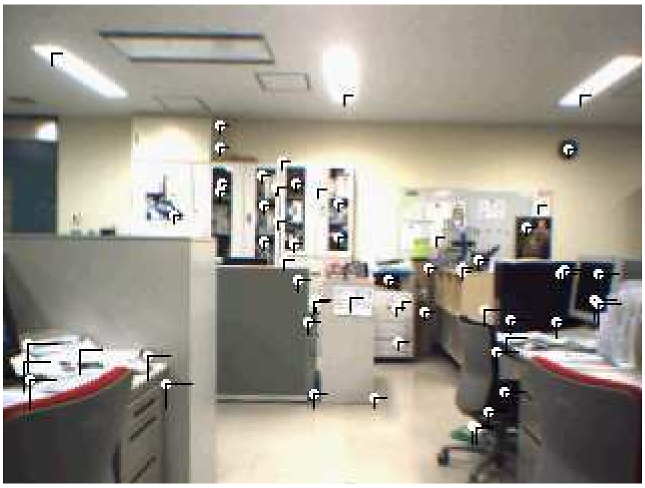


Figure 1 Trinocular correspondences: The white filled circles are the stable keypoints, the black lines from the stable keypoints are horizontal and vertical disparities.

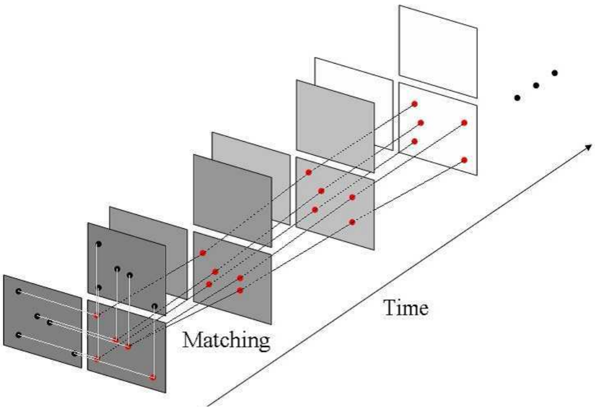


Fig. 2 Tracking keypoints

unique correspondence and the different of the vertical and horizontal disparities less than 20% of each other. Each stable keypoint has 3-D coordinate $[X\ Y\ Z]^{\mathrm{T}}$, disparity d (mean of vertical and horizontal disparities) and the SIFT keypoint. The keypoint was extracted from the right image. Fig. 1 shows the stable keypoints and their depths.

B. Tracking of Keypoints

Temporal correspondences of the stable keypoints are made between two successive frames by using the right image of the trinocular images as Fig. 2. When n keypoints are extracted at time t and m keypoints are extracted at time t-1, if keypoints $i(=1,2,\cdots,n)$ and $j(=1,2,\cdots,m)$ satisfy the following conditions:

$$\frac{\left|\mathbf{Y}_i(t) - \mathbf{Y}_j(t-1)\right|}{\mathbf{W}_{yy}(t) + \mathbf{W}_{yy}(t-1)} \le 3$$

$$\left|x_i(t) - x_j(t-1)\right| \le 40$$

$$\left|y_i(t) - y_j(t-1)\right| \le 40$$

$$\frac{\left|s_i(t) - s_j(t-1)\right|}{s_i(t)} \le 0.2$$

$$\frac{\left|s_i(t) - s_j(t-1)\right|}{s_j(t-1)} \le 0.2$$

$$\frac{\left|d_i(t) - d_j(t-1)\right|}{d_i(t)} \le 0.2$$

$$\frac{\left|d_i(t) - d_j(t-1)\right|}{d_j(t-1)} \le 0.2$$

$$\left|o_i(t) - o_j(t-1)\right| \le 20$$

$$(\mathbf{f}i(t) - \mathbf{f}j(t-1))^T(\mathbf{f}i(t) - \mathbf{f}j(t-1)) \le 0.0025$$

where W is the observation error covariance matrix. Then they are corresponding candidates. Among those candidates, correspondences are decided in the same way as in Sec.2.1.

III. Estimating Egomotion and Map Building

The positions of the keypoints in the camera coordinate changes according to the robot egomotion. The egomotion of the robot is calculated by the motion of keypoints. Once the egomotion is obtained, the 3-D position of keypoints in the world coordinate is determined, and the environment map is built.

A. Estimating Egomotion

The estimated position of keypoint $\mathbf{x}(t+1)$ in the camera coordinate at time t+1 is predicted from the previous position $\mathbf{x}(t)$ as:

$$\hat{\mathbf{x}}_i(t+1) = \mathbf{R}(\theta(t))(\mathbf{x}_i(t) - \mathbf{v}(t))$$

where the 3-D position of n keypoints in the camera coordinate is $\mathbf{x}_i = [X_i\ Y_i\ Z_i]^T\ (i = 1,2,\cdots,n)$, translation vector of mapping robot is $\mathbf{v}(t)$, rotation of that is $\theta(t)$, and $\mathbf{R}(\bullet)$ is a rotation matrix.

The egomotion, $\mathbf{m}(t) = [\mathbf{v}(t)\ \theta(t)]^T$, can be estimated by minimizing the sum of Mahalanobis distance between $\mathbf{x}(t+1)$ and $\hat{\mathbf{x}}(t+1)$ as shown in the following equations:

$$\mathbf{q}_i(t) = \mathbf{x}_i(t+1) - \hat{\mathbf{x}}_i(t+1)$$

$$\mathbf{Q}_i(t) = \mathbf{S}(\mathbf{x}_i(t)) + \mathbf{S}(\mathbf{x}_i(t+1))$$

$$\mathbf{m}^*(t) = \arg\min_{\mathbf{m}(t)}\left(\sum_{i=1}^{n-1} \mathbf{q}_i(t)^T \mathbf{Q}_i(t)^{-1} \mathbf{q}_i(t)\right)$$

where $\mathbf{S}(\mathbf{x})$ is the estimated error covariance matrix of the trinocular stereo measurements $\mathbf{x}$. The possible $\mathbf{m}(t)$ is limited in the small range because time slice is small enough. In this research, the brute force search was employed for the robot egomotion. The correspondence that has the largest Mahalanobis distance does not be used to estimating egomotion.

There are some stable keypoints that were not tracked. They are transformed to the world coordinate then their correspondences are searched in the map. The egomotion is estimated again using these correspondences.

B. Update 3-D Position of Keypoints

The 3-D position of the keypoints is transformed from the camera coordinate into the world coordinate. We suppose that the keypoints on the environment do not move and deform. Kalman filter is employed for precisely estimating their positions. This filtering can be represented by the trinocular stereo measurements y, corrected 3-D position $\hat{\mathbf{x}}$ of the keypoints, and predicted position $\tilde{\mathbf{x}}$ of the keypoints as follows:

$$\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}_k + \mathbf{P}_k \mathbf{C}_k^T \mathbf{W}_k^{-1}\{\mathbf{y}_k - \mathbf{C}_k \tilde{\mathbf{x}}_k\}$$

$$\tilde{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1}$$

$$\mathbf{P}_k = (\mathbf{M}_k^{-1} + \mathbf{C}_k^T \mathbf{W}_k^{-1} \mathbf{C}_k)^{-1}$$

$$\mathbf{M}_k = \mathbf{P}_{k-1}$$

where $\mathbf{M}$ is the predicted error covariance matrix, $\mathbf{P}$ is the corrected error covariance matrix, and $\mathbf{C}$ is a transformation matrix represented as:

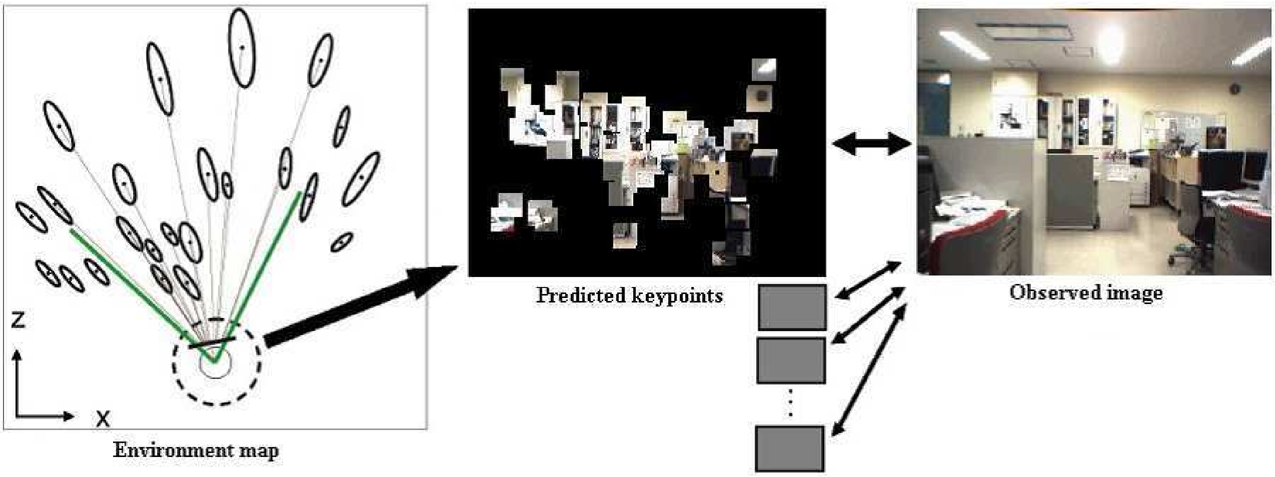


Fig. 3 The flow of self localization of working robots

The environment map is built by iterating the above process.

Finally those feature points are deleted which are observed only once. The feature of the landmarks changes according to the view direction. Then the feature vectors are registered every 10 degree. Each feature point on the map has a set of features: observation position (X_o, Y_o) and 36 SIFT keypoints.

IV. Self Localization of Working Robots with Monocular Vision

Once the robot with trinocular vision builds the environment map, the working robot self-localizes on the map from the 2-D image cues without any 3-D sensors. The robot captures an image at the current position, and extracts SIFT keypoints from the image and makes correspondence to the keypoints registered on the map. Self-localization is achieved by finding the position which minimizes the difference between predicted keypoint's positions and those in the actually observed image (look Fig. 3).

A. Prediction the 2-Dposition of the Keypoints

The position of the feature points on the map is projected by the projection. It predicts each keypoint's position in the image to be observed. We assume that the robot position is known approximately. Then the feature points are deselected that have not been observed from the direction. The center of Fig. 3 shows the position of the keypoints in the camera coordinate where for each keypoint the surrounding image is attached at the position.

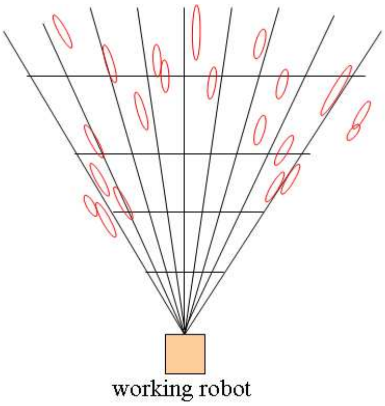


Fig.4 Reducing the Projected Points

The predicted 2-D position of n keypoints $\hat{\mathbf{u}}_i = [\hat{x}_i\ \hat{y}_i]^T\ (i = 1,2,\cdots,n)$ in the image coordinate is determined from the 3-D position of keypoint $\mathbf{w}_i = [X_i\ Y_i\ Z_i]^T$ in the map as:

$$\hat{\mathbf{u}}_i = \mathbf{P}(\mathbf{R}(\phi)(\mathbf{w}_i - h))$$

where h and ϕ are position and direction of the robot respectively, $\mathbf{P}(\bullet)$ is the projection of the 3-D position.

All the viewable points are too much to localize the position in real time. Then the projected feature points should be reduced. Nearer points have advantage to obtain translation, and farther points have advantage to obtain rotation. We segment the feature points based on view direction and distance from robot as Fig. 4, and pick up a distinctive feature point in each region. The distinctive feature is defined by the highest observation frequency and the highest contrast.

B. Matching and Self-Localization

Let the 2-D position of m keypoints in the input image be denote by $\mathbf{u}_j = [x_j\ y_j]^T$ (j = 1,2,…,m), corresponding pair candidates should satisfy the following constraints:

$$|x_i(t) - x_j(t-1)| \leq 15$$

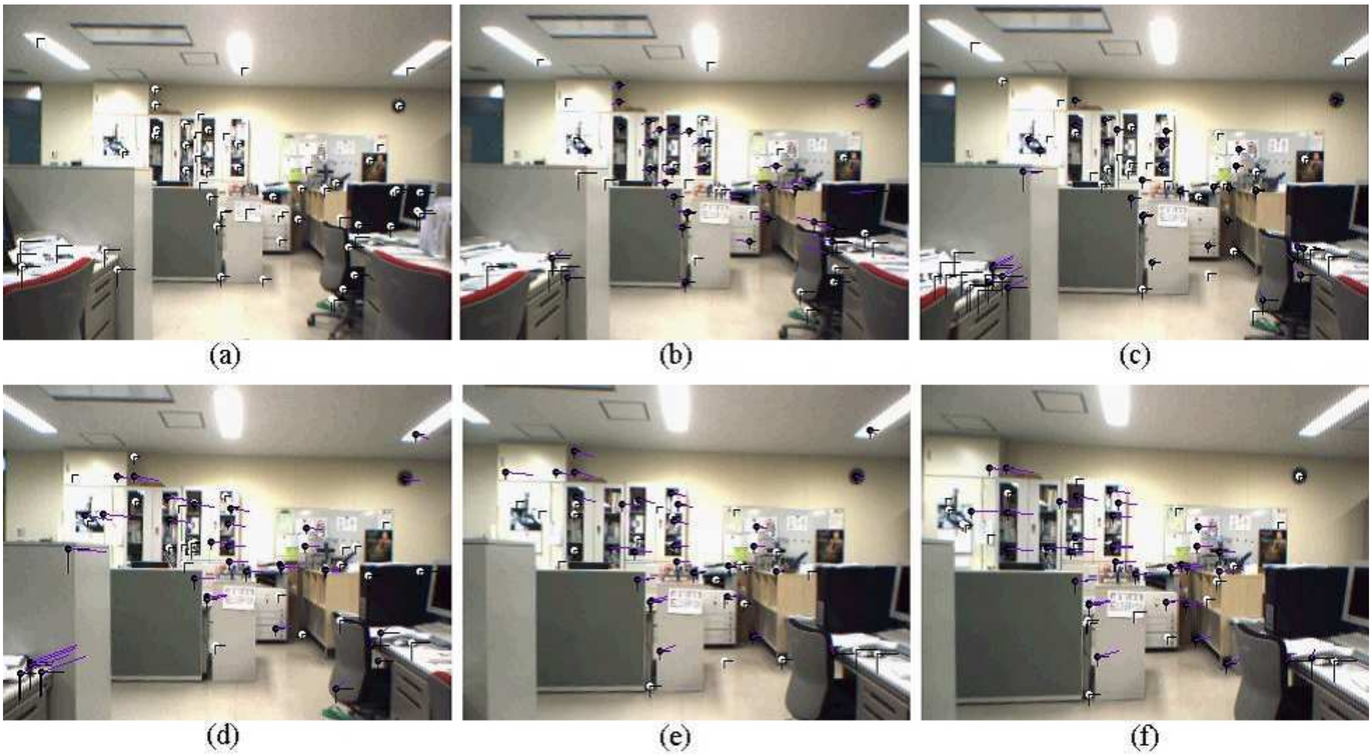
$$|y_i(t) - y_j(t-1)| \leq 15$$

$$|o_i(t) - o_j(t-1)| \leq 20$$

$$(\mathbf{f}i(t) - \mathbf{f}j(t-1))^T (\mathbf{f}i(t) - \mathbf{f}j(t-1)) \leq 0.0025$$

Among those candidates, correspondences are decided as in the case of Sec.2.1.

The robot position p is estimated by minimizing the mean square distance between the actual and predicted position:



(a) (b) (c)

(d) (e) (f)

Fig.5 keypoint tracking result: The black filled circles represent the keypoints matching to those of the previous frame and the white filled circles represent ones that no matching keypoint is found. The lines from the black filled circles represent the optical flow of the keypoints (their tail ends show the positions of the corresponding keypoints at the previous frame).

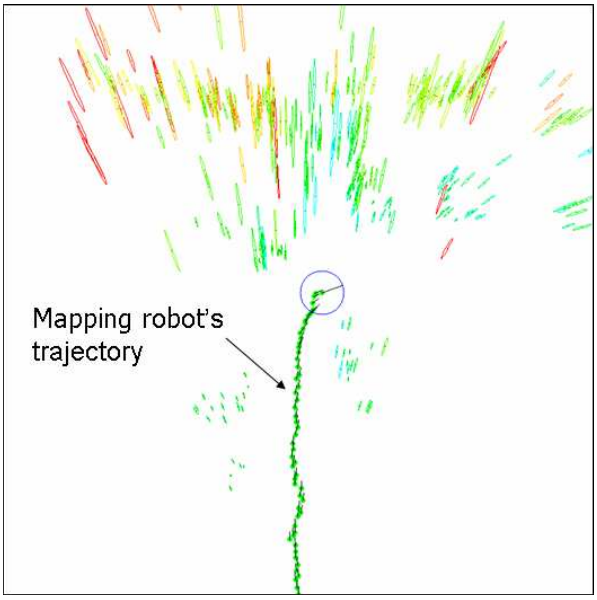


Fig.6 Keypoint map: The map scale is 1 cm per pixel. The origin of the map coordinate is located at the ini- tial robot position, the center bottom of the map. The estimated robot positions (the right camera position) and its moving trajectory are shown as the filled small circle and the solid lines. The short black solid lines on the filled circle represent the viewing directions of the robot. The large non-filled circle on the trajectory tail represents the robot size (approximately 600mm). The 430 keypoints observed at least twice is drawn in the map as the covariance elipsoids.

$$\mathbf{p}^* = \arg\min_{\mathbf{p}} \left(\frac{\sum_{l=1}^{k} (\mathbf{u}_l - \hat{\mathbf{u}}_l)^2}{k} \right)$$

where k is the number of the corresponding keypoints.

V. Experimental Result

We show the experimental results of the keypoint map generation in an indoor scene and the self-localization of the working robot. We implemented the mapping robot with a omni directional mobile robot, the trinocular camera (99.6 mm baselines, 320x240 image resolution and 28 fps sampling rate) and a desktop PC (Pentium4 2.80GHz, memory 1GB, OS Windows XP Professional Version 2002 SP2). First we employed the mapping robot and built a keypoint map of an office room. Then we re-employed the robot enabled with only one camera of the trinocular device as the working robot in order to simulate the localization by monocular vision.

A. Matching and Self-Localization

Fig.s 5(a)-(f) show the keypoint extraction results of the first 6 frames of an image sequence for map building. The number of extracted stable keypoints from the one frame of the image sequence was 51 at least, 97 at most, and 67 in average.

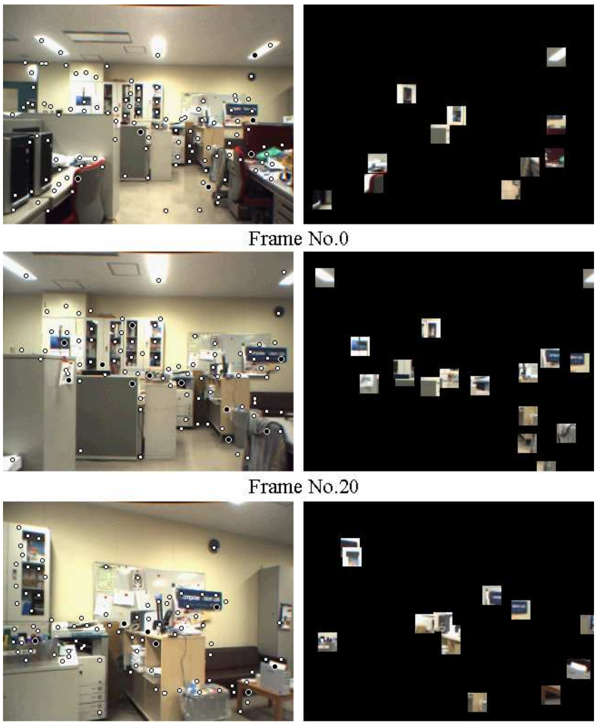
The egomotion between successive image frames were estimated by a brute force search in the following range:

TABLE 1
REDUCED POINTS

	x[mm]	z[mm]	φ[deg]
Mean error	9.9	-14.3	0.2
Variance	2294.2	5419.8	0.8
Deviation	**47.9**	**73.6**	**0.9**

TABLE 2
ALL POINTS

	x[mm]	z[mm]	φ[deg]
Mean error	-3.1	26.8	0.2
Variance	3026.8	2714.1	1.2
Deviation	**55.0**	**52.1**	**1.1**



Frame No.0

Frame No.20

Frame No.40

Fig.7 Keypoint matching: The left column shows input images, the right column shows reconstructed scene appearance. The black filled circles represent the keypoints corresponding to the projected key points in the map, the white filled ones are those not corresponding to them.

$$-100 \le \Delta X \le 100 \quad \textit{every}\ 10[mm]$$
$$-10 \le \Delta Z \le 150 \quad \textit{every}\ 10[mm]$$
$$-10 \le \Delta \theta \le 10 \quad \textit{every}\ 1[\mathrm{deg}]$$

Fig. 6 shows the generated keypoint map. It can be seen that the covariances of the registered key points near the robot trajectory or those observed from various directions are small enough.

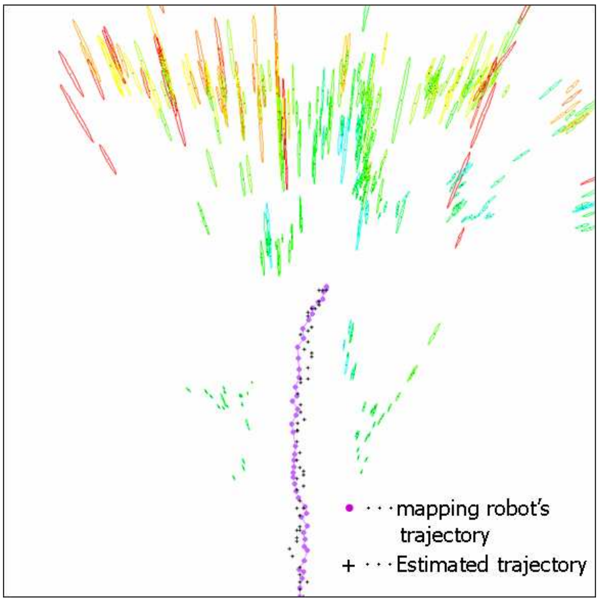


Fig.8 Estimation trajectory of working robot: The black crosses represent the estimated trajectory of the working robot's position.

B. Self-localization results using Keypoint Map

The initial robot position at the first frame was given approximately. For the following frames, the estimation result at the previous frame was given as the initial estimate. The range of the search is set as follows:

$$-100 \le \Delta X \le 100 \quad \textit{every}\ 50[mm]$$
$$0 \le \Delta Z \le 150 \quad \textit{every}\ 50[mm]$$
$$-10 \le \Delta \theta \le 10 \quad \textit{every}\ 2[\mathrm{deg}]$$

Since the mapping robot's left camera trajectory is known, the localization accuracy can be verified using the left camera images used in the map building process. Table 1 and Table 2 compare the localization accuracy with reduced points to the accuracy with all points. They have similar deviation and their required time are 1.73 [sec/frame] and 14.30 [sec/frame]. Reducing points is able to reduce the required time about 10%. The mean ratios of the number of the projected points to the number of the matched points are 63.51% and 32.37%.

Fig. 7 shows the input images captured by the working robot (the left column) and the corresponding scene models obtained as the result of the keypoint matching between the image and the map (the right column) with reducing points.

Fig. 8 shows the self-localization results of the working robot by monocular vision. The working robot moved near the trajectory of the mapping robot by human guidance. The estimated trajectory of the working robot is close to the trajectory of the mapping robot.

VI. CONCLUSION

We have presented a SIFT-based mapping method and self-localization method for mobile robots. Once a mapping robot with the trinocular camera builds the map for an unknown environment, the working robots with only one 2-D camera can localize own position by matching the visible SIFT keypoints to those on the map. The current localization method using a simple brute force matching is much time-consuming. The keypoint matching needs more rapid algorithm like the Best-Bin-First [5], and the position searching also needs like the steepest descent method.

The mapping robot should have a strategy where to explore and what kind of keypoints to select and store into the map for the accurate navigation. The working robot should also have a strategy where to move for no missing of its location. The developments of those strategies are the future works.

ACKNOWLEDGEMENT

We acknowledge Satoshi Yamaguchi and Yutaka Yamada for many helpful discussions during the development.

REFERENCES

[1] Stephan Se, David Lowe, Jim Little, “Mobile Robot Localization and Mapping with Uncertainty Using Scale-Invariant Visual Landmarks”, *International Journal of Robotics Research*, Vol. 21, No. 8, 2002, pp. 735-758.

[2] Yoshiro Negishi, Jun Miura, Yoshiaki Shirai, “Map Generation of a Mobile Robot by Integrating Omnidirectional Stereo and Laser Range Finder”, *Journal of the Robotics Society of Japan*, Vol. 21, No.6, 2003, pp. 690-696.

[3] Xu G., S. Tsuji, M Asada, “A Motion Stereo Method Based on Coarse to Fine Control Strategy”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 9, No. 2, 1987, pp. 332-336

[4] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton and Olivier Stasse, “MonoSLAM: Real-Time Single Camera SLAM”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 6, 2007, pp. 1052-1067.

[5] David G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints”, *publication in the International Journal of Computer Vision*, 2004.

[6] Jeffrey S. Beis and David G. Lowe, “Shape indexing using approximate nearest-neighbour search in highdimensional spaces”, *In Conference on Computer Vision and Pattern Recognition*, Puerto Rico, pp. 1000-1006, 1997.