This is an electronic version of an article published in ADVANCED ROBOTICS, [Volume 26, No.3-4, pp-327-348, 2012].
ADVANCED ROBOTICS is available online at:
www.tandfonline.com/10.1163/156855311X614608.

Outdoor Localization using Stereo Vision under Various Illumination Conditions

Kiyoshi Irie Tomoaki Yoshida Masahiro Tomono

Future Robotics Technology Center, Chiba Institute of Technology,
2-17-1 Tsudanuma, Narashino-shi, Chiba, Japan, irie@furo.org

Abstract

We present a mobile robot localization method using a stereo camera only. Vision-based localization in outdoor environments is still challenging issue because of large illumination changes. To cope with varying illumination conditions, we use 2D occupancy grid maps generated from 3D point clouds obtained by a stereo camera. Furthermore, we incorporate salient line segments extracted from the ground into the grid maps. The grid maps are not much affected by illumination conditions because occupancy information and salient line segments can be robustly obtained. On the grid maps, the robot poses are estimated using a particle filter that combines visual odometry and map-matching. We use edge point based stereo SLAM to obtain occupancy information and robot ego-motion estimation simultaneously. We tested our method under various illumination and weather conditions including sunny and rainy days. The experimental results showed the effectiveness and robustness of the proposed method. Our method enables localization under extremely poor illumination conditions which are too challenging for existing state-of-the-art methods.

Keywords: localization, mobile robot, stereo vision

1 Introduction

Outdoor navigation is an important issue in mobile robotics and localization is one of the essential components of navigation. Localization for outdoor navigation has been studied extensively and there have been proposed many methods combining inertial sensors such as odometry, and external sensors such as GPS, laser scanners and cameras [1][2][3].

Employing stereo vision is one of the promising approaches to mobile robot localization. Stereo cameras can obtain 3D range data at high frame rate and also capture colors and textures, which cannot be detected sufficiently by laser scanners. Recently, image features with distinctive local descriptors, such as Sale Invariant Feature Transform (SIFT) [4], have been employed for mobile robot localization [5]. These image features are useful to identify landmarks in indoor environments without large illumination

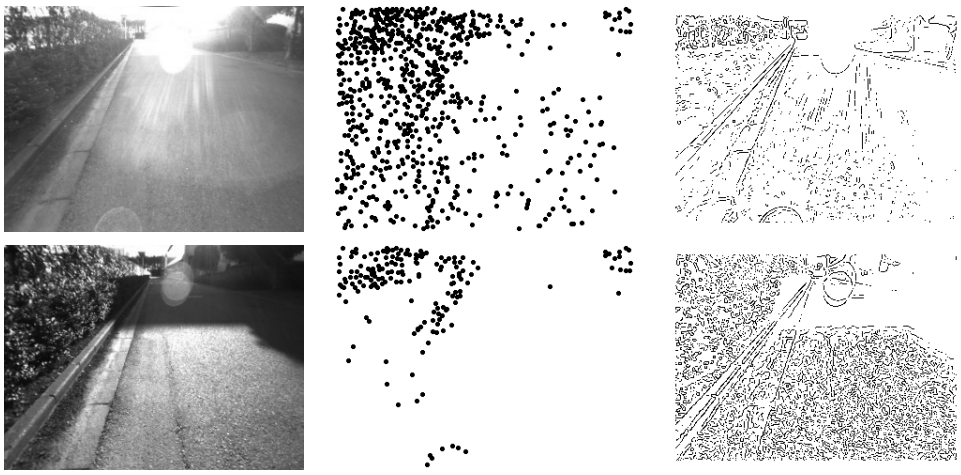


Figure 1: Two images of the same place captured at different times and result of image feature detection. Left: original image. Center: Key points detected by SIFT. Right: Detected edge points. Lowe's SIFT Keypoint Detector [6] was used to detect the SIFT key points.

changes. However, drastic illumination changes, which often occur in outdoor environments, make it difficult to obtain stable image features.

Fig. 1 shows examples of image feature extraction in outdoor environments. Two images of the same place which were captured at different times and SIFT key points are detected in each image. As can be seen, the results of the SIFT key point detector vary depending on illumination conditions. Additionally, none of the detected key points was matched between two images using the SIFT key point matcher. Similar results were obtained for edge points as shown in the right column of the figure. Thus, so far it would be difficult to implement robust outdoor localization using image features only.

This paper proposes a localization method using a stereo camera which overcomes the problems mentioned above. The proposed method estimates the robot motion by visual odometry and corrects its accumulated errors using a map-matching algorithm, which is based on the shapes of 3D point clouds obtained by the stereo camera. The map-matching is actually performed using 2D grid maps generated by projecting 3D point clouds onto the ground. The projected 2D grid maps are stable under various illumination conditions and also are less computationally expensive than 3D maps. For environments without 3D features, such as wide roads and open spaces, we extract salient line segments from the ground surface and incorporate them into the grid map as additional landmarks. We refer to them as *road landmarks* in this paper. We employ a particle filter which fuses visual odometry and map-matching to estimate the position of the robot. Our method can be implemented with only a stereo camera; no motion sensors or odometry are required. However, other sensors such as wheel odometry and gyroscope, can be used to improve localization accuracy.

Our method combines 3D range data and image features in an effective manner to enhance robustness to illumination changes. The shapes of the 2D grid maps generated from 3D point clouds are not much affected by illumination conditions. Salient line segments on the ground can be extracted stably under

various illumination conditions. In urban environments, there are plenty of 3D features including walls, curbs, bushes, and trees. Also, a number of line segments, such as road boundaries and traffic signs, can be found on the ground. Thus, our method is applicable to many man-made outdoor environments. We found that our method was successfully performed in experiments on paved roads and tiled pedestrian areas under various weather conditions.

The reminder of this paper is organized as follows. After presenting related work in section 2, we present our method in sections 3 and 4. Experiments under various illumination conditions are presented in section 5. Discussion is presented in section 6 with comparison to existing methods.

2 Related Work

Vision-based outdoor navigation has been studied for decades [7]. Many methods of ego-motion estimation using vision, such as visual odometry, have been proposed [8][9] but visual odometry is not sufficient because errors accumulate over time. To correct the accumulated errors, landmark-based localization is necessary. Many features and objects have been used as landmarks for outdoor navigation; road boundary detection for autonomous driving [10], buildings [11] and Braille blocks [12].

Royer et al. used a single camera and structure-from-motion approach without odometry [13]. However, in their method, the scale is given manually because it cannot be determined by a monocular vision. Agrawal et al. presented a localization system based on a visual odometry using a stereo vision, while IMU and GPS are required to correct the error in the visual odometry [14].

Some navigation methods do not use explicit landmarks [15][16]. In these methods, the robot navigates along a pre-learned path given as an image sequence, but precise robot positions on the map cannot be obtained.

Recently, appearance-based localization methods which are robust to changes in lighting have been proposed [17][18]. These methods do not provide precise localization on a metric map since they provide only topological mapping and localization.

In contrast to the above approaches, our method requires only a stereo camera. Since our method uses grid maps containing both 3D shapes and image features, we consider it applicable to structured environments as well as to less-structured environments such as passage without apparent road boundaries and open spaces without 3D features.

3 Grid Map Generation

The procedure of generating grid maps is the most crucial part of our method since our localization is based on map-matching. A global map is built in advance and local maps are generated and matched on-line to estimate the pose of the robot. The local and global grid maps contain both occupancy information and salient line segments on the ground. Each cell in a grid map is labeled as *occupied*, *free*, *road landmark* or *unknown*.

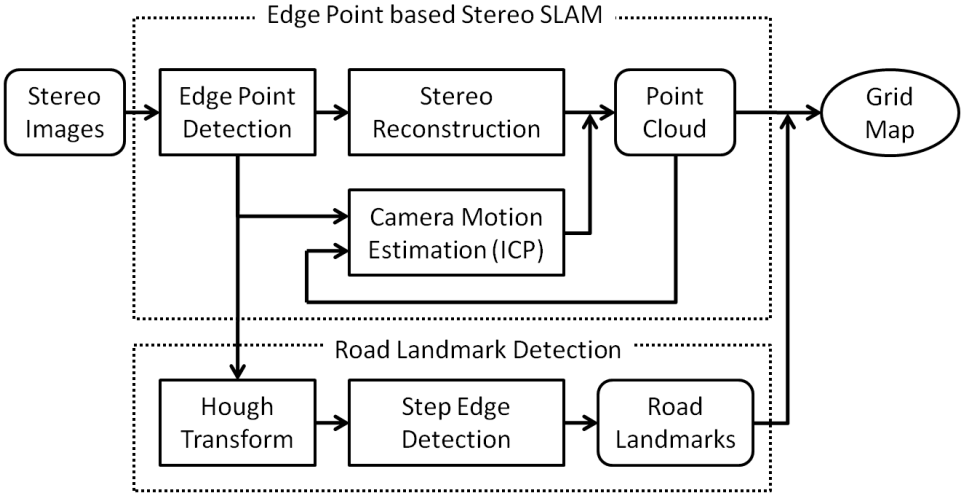


Figure 2: Procedure of grid map generation

In global map generation, we manually navigate a robot in the target environment to collect a stereo image sequence. The collected stereo images are then processed off-line to build a map in the following steps:

1. Create a 3D point cloud from stereo images and simultaneously estimate the trajectory of the robot. We use edge point based ICP to simultaneously estimate camera motion and build a 3D point cloud map.

2. Project the 3D point cloud onto a 2D grid map and label each cell as occupied or free according to the height of the points in the cell.

3. Extract salient line features on the ground from the images and label the cells that contain the line features as road landmarks. Hough transformation and step edge detector are used to extract line segments.

4. Close the loop based on 2D graph-based SLAM

Fig. 2 illustrates the map generation procedure.

3.1 Obtaining Point Cloud using 3D Stereo SLAM

A 3D point cloud map is built by edge point based stereo SLAM method [19]. The method uses image edge points which are detected from not only long segments but also fine textures.

We refer to a pair of left and right images as *stereo frame* (*frame*, for short). The 3D edge point $P_c = (X, Y, Z)^T$ is calculated from point $(x_l, y_l)^T$ on the left image and point $(x_r, y_r)^T$ on the right image based on the parallel stereo formula.

The camera motion from time $t-1$ to t is estimated by matching the 3D points reconstructed from frame I_{t-1} with the 2D points detected in frame I_t. The registration is performed using a variant of

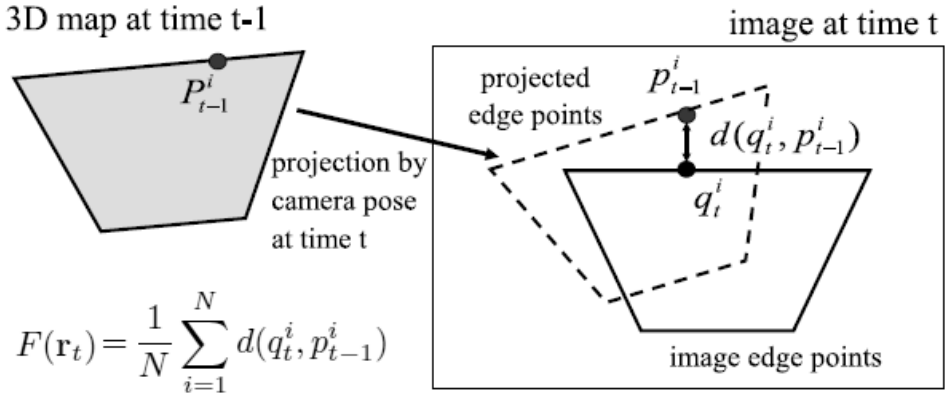


Figure 3: The 3D-2D ICP algorithm minimizes the cost function F between projected point $p_t^i - 1$ and image edge point q_t^i

ICP algorithm on the image plane as illustrated in Fig. 3. Let $\mathbf{r}_t$ be the camera pose at t, P_{t-1}^i be the i-th 3D edge point reconstructed at $t-1$, and p_{t-1}^i be the projected point of P_{t-1}^i onto image I_t. Let q_t^i be the image edge point at t, which corresponds to p_{t-1}^i. A cost function F is defined as follows:

$$F(\mathbf{r}_t) \quad = \quad \frac{1}{N}\sum_{i=1}^{N} d(q_t^i, p_{t-1}^i) \tag{1}$$

Here, $d(q_t^i, p_{t-1}^i)$ is the perpendicular distance between p_{t-1}^i and the edge segment on which q_t^i lies.

Camera motion $\mathbf{r}_t$ and edge point correspondences are searched by minimizing $F(\mathbf{r}_t)$ using the ICP algorithm. The initial value of $\mathbf{r}_t$ is set to $\mathbf{r}_{t-1}$, and the initial correspondence q_t^i of p_{t-1}^i is set to the edge point that is the closest to p_{t-1}^i in terms of Euclidean distance. By repeating the minimization of $F(\mathbf{r}_t)$ and edge point matching, the optimal $\mathbf{r}_t$ and edge point correspondences are obtained. A robust cost function [20] is employed to cope with outliers.

Based on the obtained camera pose $\mathbf{r}_t$, a 3D map is built by transforming the intra-frame 3D points from the camera coordinate system to the world coordinate system. Only the 3D points tracked for more than n_1 frames (typically $n_1 = 4$) are added to the 3D map. Also, 3D points with large variance are removed. This filter is useful to eliminate blurred edges and moving objects. The advantage of using edge points is that the number of edge points detected is usually much larger than other local features, and it is favorable for the purpose of building occupancy maps. In typical urban outdoor environments, thousands of edge points are detected from one QVGA (320×240) image, while hundreds of keypoints are detected by the SIFT detector. An example of a point cloud map built by this method is shown in Fig. 4. Our method using edge points can obtain much denser point cloud than the one with SIFT keypoints.

The procedure described in this section is also used in visual odometry, as described in section 4.1.

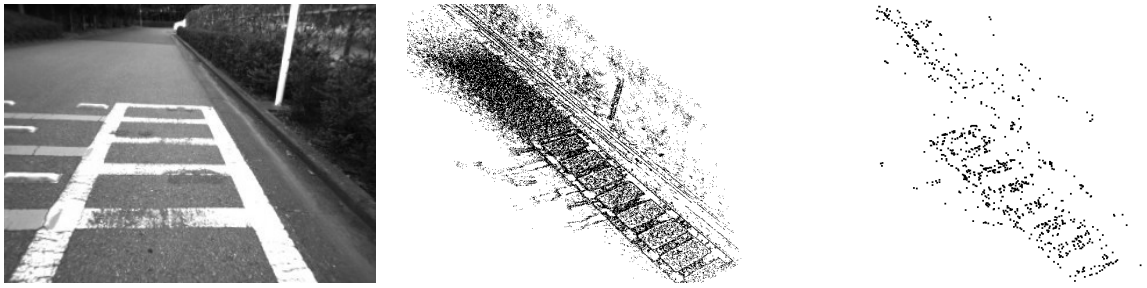


Figure 4: Example of point cloud generation. Left: One of input images. Center: Point cloud map built using edge points. Right: Point cloud build using SIFT keypoints.

3.2 Generating 2D Occupancy Grid Maps

A 2D occupancy grid map is generated by projecting the 3D point cloud map onto the ground. The ground plane is divided into square grid cells and the 3D points in the point cloud are projected onto them. To reduce the noise caused by errors in stereo matching, the grid cells are labeled as occupied or free according to the number of the contained 3D points that are higher than th_1. In our implementation, the size of the cells was $10cm$ and $th_1 = 15cm$.

The 6-DOF camera trajectory estimated by 3D SLAM has accumulated errors. When the camera moves long distance, accumulated errors can be large in the height direction, which increase spuriously-labeled cells in the 2D grid map. To address this problem, we make the camera height constant on the assumption that the robot moves on a flat ground. 3D points are rearranged on the ground plane based on the robot's 3-DOF poses and the camera pose relative to the robot.

3.3 Detecting Road Landmarks

In contrast to indoor environments, which usually have rich 3D features such as walls and furniture, some outdoor environments have very few 3D features. Even laser scanners can be affected by this problem, and it is even worse for stereo vision which usually has a small field of view and a limited range of stereo reconstruction. For stable localization in such areas, other landmarks than 3D features are needed.

To cope with this problem, we use salient line segments on the ground surface. In urban environments, various line segments can be found on the ground, such as road boundaries, patterns in tiled floors, and traffic signs. These salient features are detected stably under various illumination conditions partly because the distance between the camera and the ground is small.

Road landmarks are detected by the following procedure. First, edge points are detected from the input image by the Canny detector [21]. Second, lines (continuous edge points) are extracted using the Hough transform. Through these two steps, not only salient long segments but also short segments, which are useless for localization, are extracted from fine textures on paved roads, boundaries between tiles of the same color, etc. Since most of these short segments are roof or valley edges, we remove them

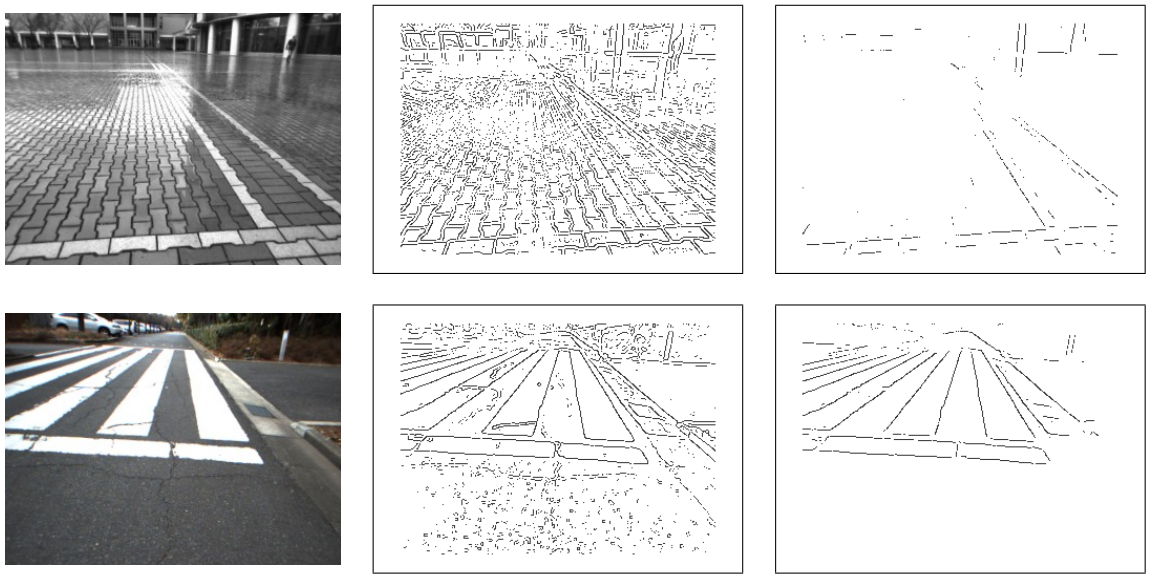


Figure 5: Detection of road landmarks. Left: original image. Center: detected edge points. Right: detected salient lines for landmarks.

using a filter that only extracts step edges. This procedure removes most of the useless tiny textures and the remaining features are useful for road landmarks. Fig. 5 shows examples of detection of road landmarks.

The extracted edge points for road landmarks are projected onto a 2D grid map. To filter out noises, only the cells that contain edge points more than a threshold are labeled as road landmarks.

3.4 Loop Closure

To reduce accumulated errors from stereo SLAM, loop closure is performed to correct the robot trajectory. We use a graph based SLAM formulation [22] with optimization, as presented in [23].

The graph is constructed as follows. When the robot moves for a certain distance, a node representing the robot pose is automatically added to the graph, and also an arc is added to represent geometric constraints between the new node and the previous node. In our current implementation, an arc to close the loop by connecting the nodes of the same place is created manually.

4 Monte Carlo Localization

Our localization method uses a particle filter based on Monte Carlo Localization [24]. In the prediction step of the particle filter, we draw a set of particles based on the robot motion estimated by visual odometry. The robot pose is denoted by $\mathbf{x} = (x, y, \theta)$, assuming the robot moves on a 2D ground plane. In the update step, the particles are weighted by the map-matching score and re-sampled according to the weights.

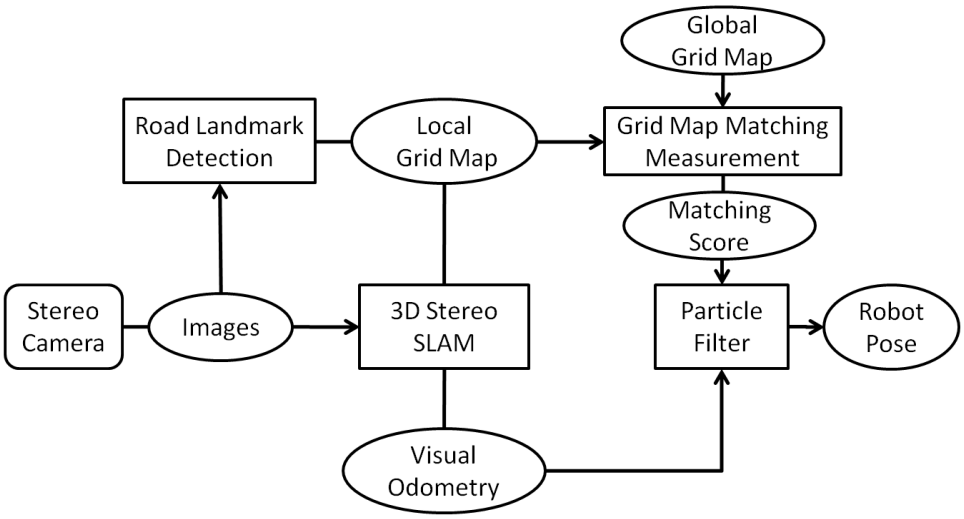


Figure 6: Procedure of proposed localization

4.1 Visual Odometry

3-DOF robot motion is calculated by firstly estimating 6-DOF camera motion using visual odometry and then projecting the motion onto the ground plane. The 6-DOF camera motion estimation is done basically in the same manner with the stereo SLAM described in section 3.1. The difference is that no global maps are generated by the visual odometry, to reduce memory consumption. The visual odometry uses local point cloud maps to estimate camera motion. A local map is created by integrating 3D points from multiple frames since 3D points reconstructed from one stereo frame can have large errors. The local maps created in this procedure are re-used in grid map matching described in section 4.3.

4.2 Prediction Step

In the prediction step, the particle filter uses 3-DOF robot motion $\mathbf{u}_t = (\Delta x_t, \Delta y_t, \Delta\theta_t)^T$, which is calculated by projecting the 6-DOF camera motion according to Eq.(3). Here, T_{camera}^{robot} is the transformation from the camera coordinate system to the robot coordinate system. $\mathbf{x'_t} = (x, y, z, \phi, \theta, \psi)^T$ is the 6-DOF robot pose (ϕ: roll, θ: pitch, ψ: yaw), and $\mathbf{r_t}$ is the 6-DOF camera pose. $\Delta\mathbf{x'_t}$ is the relative pose of $\mathbf{x'_t}$ with respect to $\mathbf{x'_{t-1}}$.

$$\mathbf{x'_t} = T_{camera}^{robot}\mathbf{r_t} \tag{2}$$

$$\mathbf{u_t} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \Delta\mathbf{x'_t} \tag{3}$$

Approximating the error by a normal distribution, the robot pose represented by i-th particle $\mathbf{x}_t^i = (x_t^i, y_t^i, \theta_t^i)^T$ is calculated by using Eq.(4).

$$\mathbf{x}_t^i = \mathbf{x}_{t-1}^i + R(\theta_{t-1}^i)(\mathbf{u}_t + \mathbf{w}_t^i) \tag{4}$$

$$R(\theta) = \begin{pmatrix} cos(\theta) & sin(\theta) & 0 \\ -sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{5}$$

$$\mathbf{w}_t^i \sim N(0, \Sigma_t) \tag{6}$$

The covariance matrix Σ_t is determined experimentally.

4.3 Update Step

The update step of the particle filter is based on a map-matching between a global 2D grid map M_{global} and a local 2D grid map $M_{local,t}$. The global 2D grid map is built as described in section 3. The local 2D grid map is built by the same procedure, using a point cloud generated through the camera motion estimation (as described in section 4.1).

In the update step, particles are re-sampled according to the weight w^i proportional to the likelihood of the measurement as Eq. (7).

$$w^i \propto p(M_{local,t}|\mathbf{x}_t^i, M_{global}) \tag{7}$$

In our implementation, $p(M_{local,t}|\mathbf{x}_t^i, M_{global})$ is approximated by cosine correlation between the local grid map and the global grid map. Let o_k^l be the occupancy value of the k-th cell in the local grid map, and r_k^l be the road landmark value of the cell (Eq. (8) and (9)). Local map vector $\mathbf{m}_{local,t}$ is defined as Eq. (10).

$$o_k^l = \begin{cases} 1 & \text{(occupied)} \\ 0 & \text{(not occupied)} \end{cases} \tag{8}$$

$$r_k^l = \begin{cases} 1 & \text{(road landmark)} \\ 0 & \text{(not road ladnamrk)} \end{cases} \tag{9}$$

$$\mathbf{m}_{local,t} = (o_1^l, r_1^l, o_2^l, r_2^l, ..., o_N^l, r_N^l) \tag{10}$$

Let $o_{k,\mathbf{x}}^g$ be the occupancy value of a cell in the global grid map corresponding to the k-th cell in the local grid map when the robot is at $\mathbf{x}$ (and $r_{k,\mathbf{x}}^g$ is defined similarly). Global map vector $\mathbf{m}_{global,\mathbf{x}}$ is defined as Eq. (11).

$$\mathbf{m}_{global,\mathbf{x}} = (o_{1,\mathbf{x}}^g, r_{1,\mathbf{x}}^g, o_{2,\mathbf{x}}^g, r_{2,\mathbf{x}}^g, ..., o_{N,\mathbf{x}}^g, r_{N,\mathbf{x}}^g) \tag{11}$$

The cosine correlation between the local grid map and global grid map for the i-th particle is calculated as Eq. (12). The weight w^i is calculated by normalizing the correlation ρ^i as Eq. (13).

$$\rho^i = \frac{\mathbf{m}_{local,t} \cdot \mathbf{m}_{global,\mathbf{x}_t^i}}{\|\mathbf{m}_{local,t}\| \left\|\mathbf{m}_{global,\mathbf{x}_t^i}\right\|} \tag{12}$$

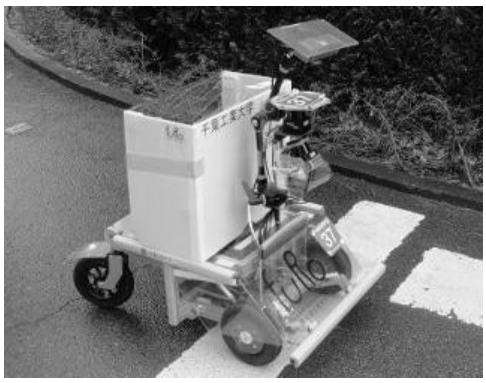


Figure 7: Robot used in experiments.

$$w^i = \rho_i / \sum_j \rho_j \tag{13}$$

4.4 Error Recovery

Although our visual odometry works well under various illumination conditions, it can fail under extremely adverse conditions. For example, direct sunlight can saturate a large part of the captured image to black or white due to the limited dynamic range of the camera. In such a case, sufficient edge points cannot be detected, which causes large errors in motion estimation.

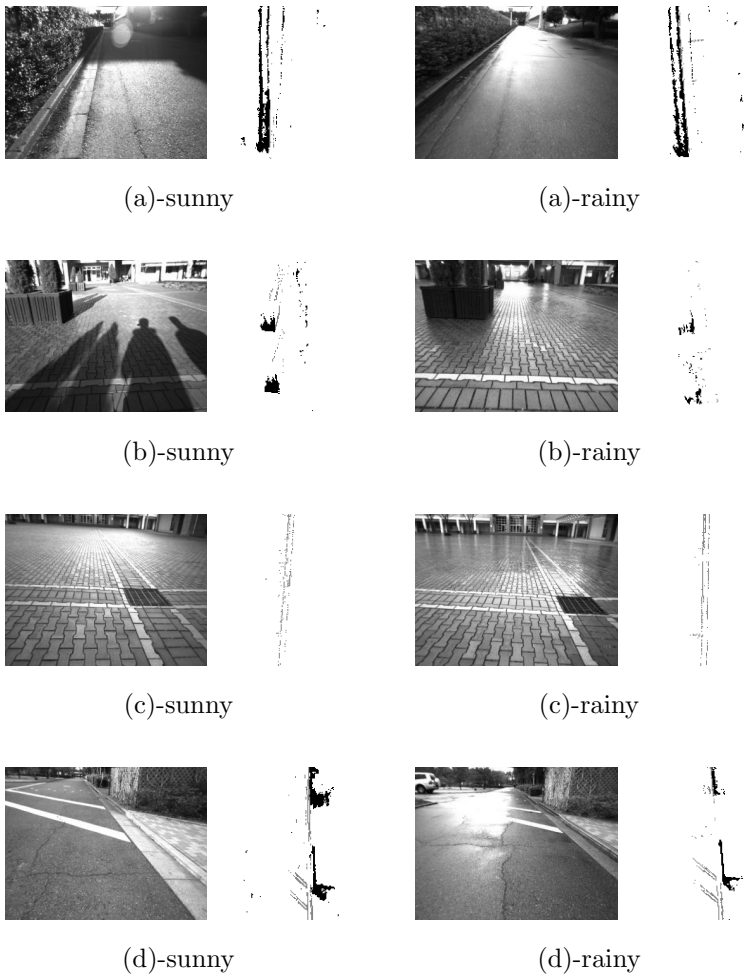
We found that this problem is similar to slip of the wheels in the case of wheel odometry, and considered it as a kind of kidnapped robot problem. Several methods have been proposed for the kidnapped robot problem [25][26]. Our solution is similar to Expanding Reset method described in [27], which is suitable when the distance of kidnap is relatively small.

5 Experiments

We implemented the proposed method on a wheeled mobile robot, which is equipped with a Bumblebee2 stereo camera (Point Gray Research, Inc.). The camera was mounted at a height of 86cm from the ground, tilted at a pitch angle of -21°. The image size used was QVGA.

5.1 Map Building under Various Illumination Conditions

Before localization experiments, we evaluated how our maps are affected by illumination conditions. For comparison, we built 2D grid maps of four areas under sunny and rainy weather conditions, respectively. Fig. 8 shows the images of the four areas and the maps built from them. A lens flare seen in (a)-sunny did not affect the map. The shadow of the building in this image was not detected as a road landmark since the shadow boundary was blurred. The shadows of several people in (b)-sunny were mostly filtered out through stereo SLAM and map generation mentioned in Section 3.1. The white lines in (c) and (d) are detected as road landmarks under sunny and rainy conditions despite light reflection by water.



(a)-sunny (a)-rainy

(b)-sunny (b)-rainy

(c)-sunny (c)-rainy

(d)-sunny (d)-rainy

Figure 8: Images and grid maps under various illumination conditions. Each row shows two sets of an image and a map of the same location obtained under sunny and rainy weather conditions. Black indicates occupied cells and grey indicates road landmark cells.

As can be seen, our map building method generates similar maps even under different illumination conditions, and this validates our map-matching approach to outdoor localization.

5.2 Localization under Various Illumination Conditions

We conducted experiments on robot localization in our campus. The first experiment was conducted on a path of $400m$, which is relatively rich in 3D features. A 2D grid map was built from an image sequence captured on a cloudy day (Fig. 9 (a)), and localization was performed on the map off-line with an image sequence captured on a sunny day (Fig. 9 (b)).

The robot was manually operated to run along the path at a maximum speed of $75cm/sec$, taking stereo images at $20fps$, 16,530 pairs of images in total. To evaluate localization accuracy, we operated the robot to pass through seven reference points, which we had determined in advance.

We compared two methods using the same data set, one using only occupancy information without

Input Data Set	Use Road Landmark	Average Error	Maximum Error
sunny	yes	$59cm$, 2.3°	$156cm$, 7.6°
sunny	no	$60cm$, 3.5°	$202cm$, 12.3°
rainy	yes	$37cm$, 2.4°	$159cm$, 6.7°
rainy	no	$98cm$, 3.0°	$277cm$, 5.4°
rainy and dark	yes	$47cm$, 2.1°	$156cm$, 4.4°
rainy and dark	no	$62cm$, 2.6°	$152cm$, 8.9°

Table 1: Localization results by proposed method

road landmarks and the other using both occupancy information and road landmarks. In the first method (without road landmarks), the average pose error at the seven reference points was $60cm$, 3.5°; the maximum pose error was $202cm$, 12.3°. In the second method (with road landmarks), the average pose error was $59cm$, 2.3°; the maximum pose error was $156cm$, 7.6°.

Fig. 10 shows experimental results by our method (with road landmarks). While the trajectory of the visual odometry is distorted by accumulated errors, our localization method successfully tracked the robot position throughout the path.

We performed the same experiment with two different data sets (rainy / rainy and dark). In all experiments, our method successfully kept track of the robot position. Table 1 shows the localization results by the proposed method. Several images captured by the robot are shown in Fig. 9.

In these experiments, we used 1,000 particles. The initial pose of the robot was given as a normal distribution with a standard deviation of $10cm$. The prediction step of the particle filter was carried out for each frame of stereo images, and the update step was executed at every 60 frames. Our implementation is partly parallelized to take advantage of multi-core processors. The processing time measured on a laptop with $2.13GHz$ Dual-Core CPU was 50 to $120ms$ for each prediction step, depending on the number of the edge points in the images, and approximately $200ms$ for each update step.

5.3 Localization in Environment with Wide Open Space

The second experiment was conducted on a path of $800m$, which includes an open space of approximately $50m \times 50m$ (Top right of Fig. 11) with few 3D features. In the open space, the 2D grid map did not have any valid occupied cells due to lack of 3D features, and only road landmarks coming from white tiles on the floor could be used for localization.

We again compared the two methods (with and without road landmarks). We collected two image sequences on a rainy day and a sunny day and one of them was in turn used to generate a global map and the other was used as the localization input. The accuracy of localization measured at 11 reference points is shown in Table 2. The method without road landmarks had significant errors in the open space and at other areas with few 3D features (see Fig. 12). By using the road landmarks, the localization

(a) cloudy (used for map)

(b) sunny

(c) rainy

(d) rainy and dark

Figure 9: Images used in the first experiment.

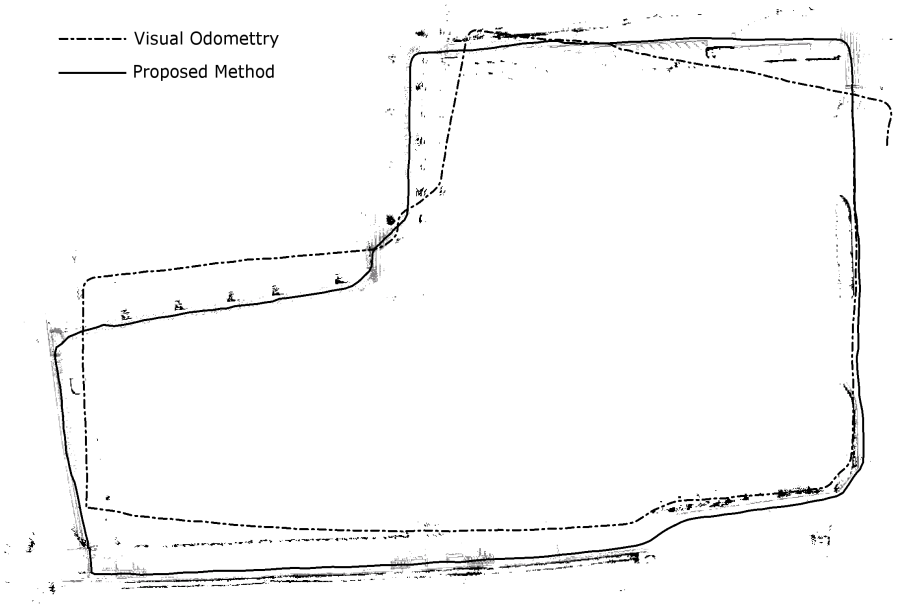


Figure 10: Localization result by proposed method and estimation by visual odometry.

Figure 11: Images captured by the robot during the second experiment.

Input Data Set	Map Data Set	Use Road Landmark	Average Error	Maximum Error
sunny	rainy	yes	41cm, 2.3°	161cm, 5.3°
sunny	rainy	no	313cm, 20.2°	670cm, 86.3°
rainy	sunny	yes	51cm, 1.9°	102cm, 5.4°
rainy	sunny	no	150cm, 3.2°	421cm, 10.1°

Table 2: Localization results in environment with a wide open space

accuracy was largely improved.

Fig. 13 shows a zoomed comparison of localization in the open space. The error ellipses were calculated approximately from particles. As can be seen in the figure, the method with road landmarks provided better estimation.

5.4 Recovery from Localization Failure

We show an example of recovery from localization failure. In an experiment on a sunny day, we found an extremely adverse condition shown in Fig. 14, in which a large part of the images was blacked out because of sunlight and shadow. As mentioned in section 4.4, visual odometry cannot estimate the motion of the robot correctly in such conditions.

We carried out an experiment with this image sequence and a map built from a rainy data set. The result is shown in Fig. 15. Visual odometry incorrectly estimated the motion of the robot for approximately 70 frames immediately before the robot turned right. After the robot finished turning to the right, localization failure is detected and expansion reset occurred, and eventually the robot was re-localized. The recovery from localization failure enables the robot to resume localization even if it encounters extremely poor illumination conditions as long as they are transient.

5.5 Localization in a Crowded Urban Environment

Finally, we tested our method in a crowded urban environment. We collected image sequences at different times on a sunny day in two courses close to Tsudanuma Station: a) Loop course including

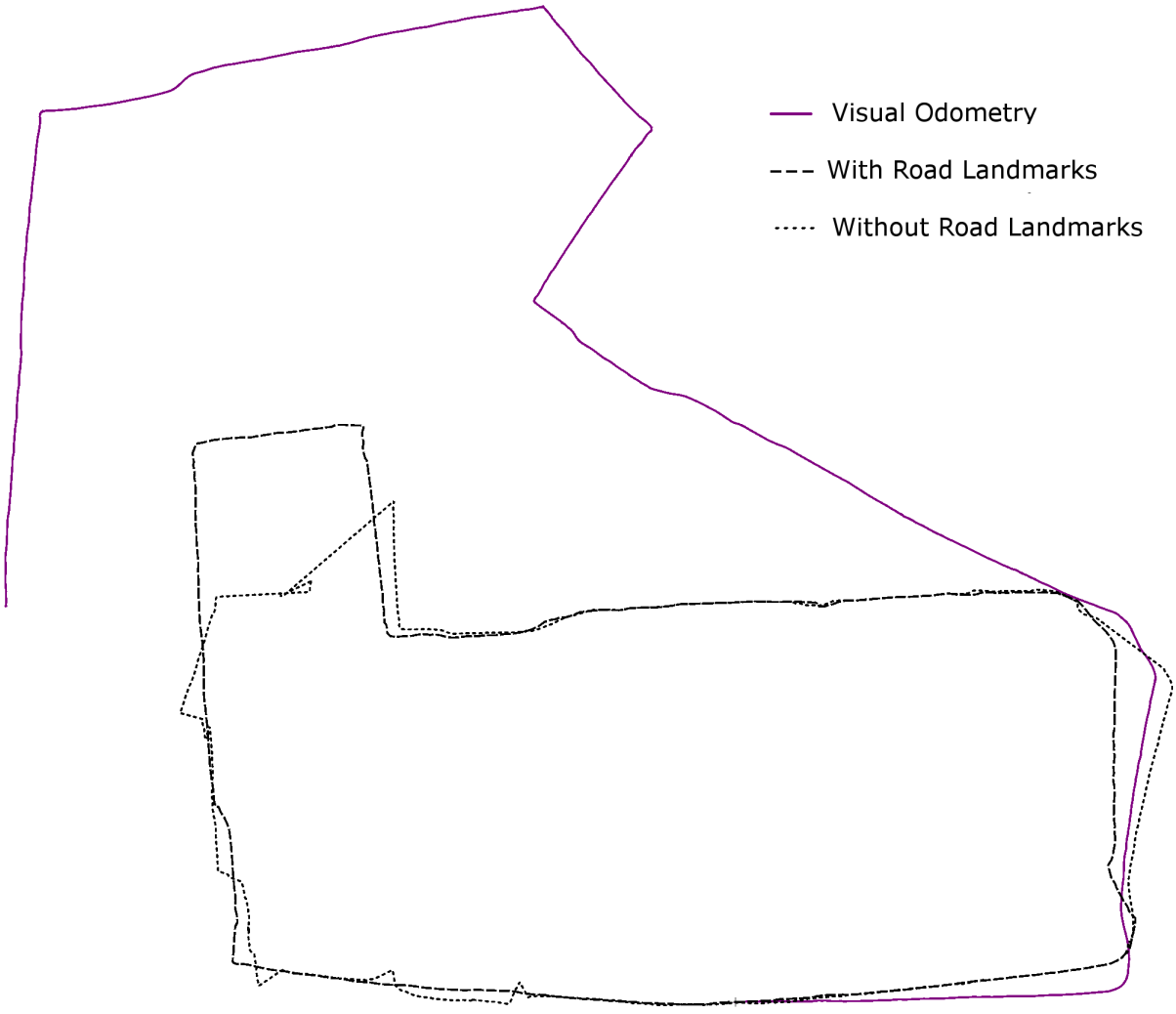


Figure 12: Trajectory obtained by proposed method (with and without road landmarks) and visual odometry.

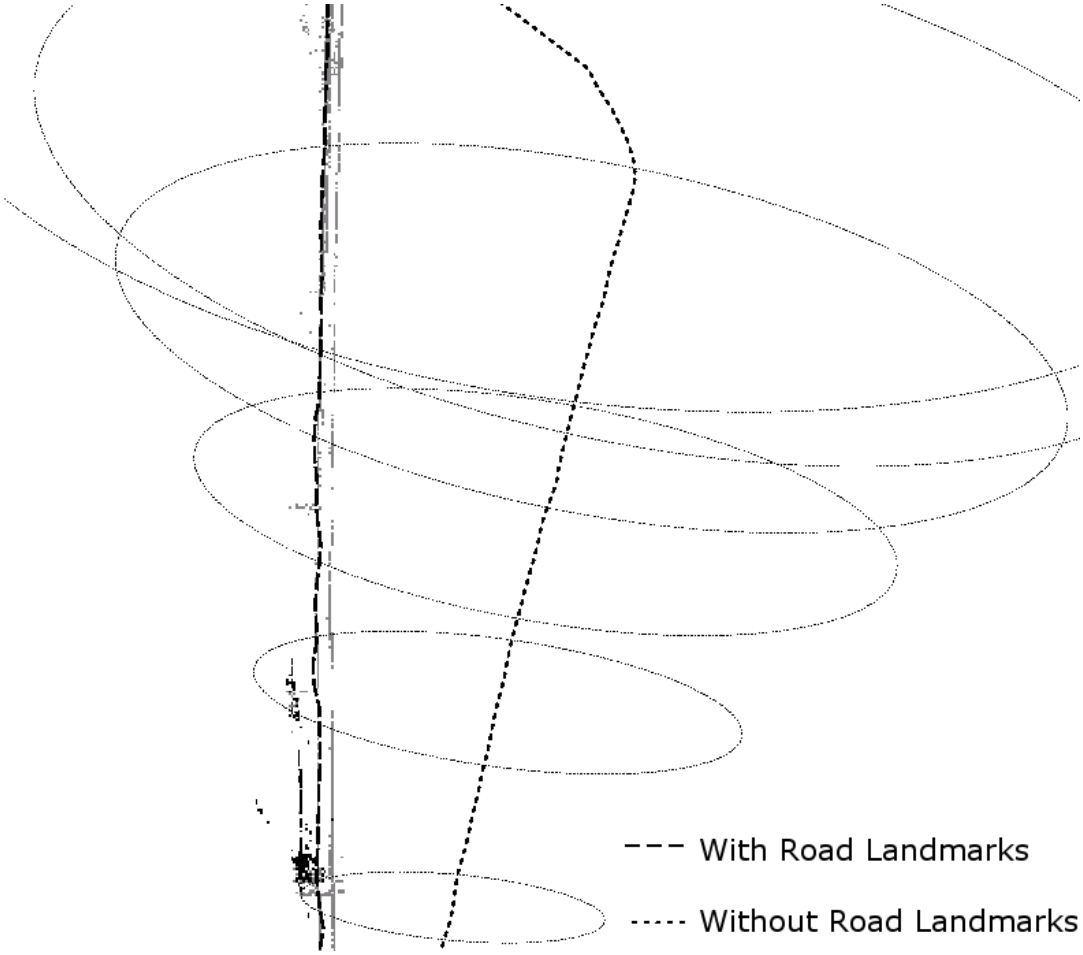


Figure 13: Localization results of two methods in an open space. The dashed line shows trajectory obtained with road landmarks. The dotted line and the ellipses show trajectory and error ellipses obtained without road landmarks.

Figure 14: Images obtained during experiment. Left: visual odometry not functional because of few edge points. Center: pole in the image used for visual odometry while turning to right. Right: camera exposure was adjusted to the shadow after turning.

Figure 15: Recovery from localization failure. Left: estimated position and error ellipses. The trajectory obtained by gyro-assisted wheel odometry is shown for comparison. The robot navigated from top- left to bottom- right in this figure. Right: distribution of particles. The triangles show particles. The red triangles show estimated position. (a) Localization is incorrect due to visual odometry failure. (b) Kidnapping is detected. (c) Robot re-localized.

Figure 16: Images collected close to Tsudanuma Station.

sidewalks and streets. b) Walkway including a pedestrian bridge. In these experiments, we used a camera mounted on a wheelchair since it was not permitted to operate a robot in the environment. As seen in Fig. 16, the dataset is very challenging because of drastic illumination changes and many dynamic objects such as pedestrians and bicycles. We generated global grid maps using two datasets (16:10 and 16:30) and found that plenty of 3D features and road landmarks can be detected in the environment.

Localization experiments were performed using the generated grid maps and the datasets of 12:30 and 12:50. The results are shown in Fig. 17. Although three failure recoveries occurred, our method kept track of the robot position without catastrophic errors throughout the course a). The causes of the failures include accumulated errors in the longitudinal direction in long straight paths, and lack of 3D features in a broad street due to the limited range of stereo reconstruction.

Our method resulted in a significant error with the course b). Extremely adverse illumination conditions hindered both visual odometry and 3D shape detection (see the second from right in Fig. 16) and our method could not recover from the failure.

6 Discussion

We have demonstrated that our method works under various illumination conditions. The robustness to illumination conditions is obtained because we use map-matching based on 3D shape information, instead of comparing image features directly. Our method works in environments with less 3D features provided that there are salient line segments on the ground. We also showed the potential for applications in real urban environments.

Nonetheless, we found some circumstances where our method is not functional. Obviously our

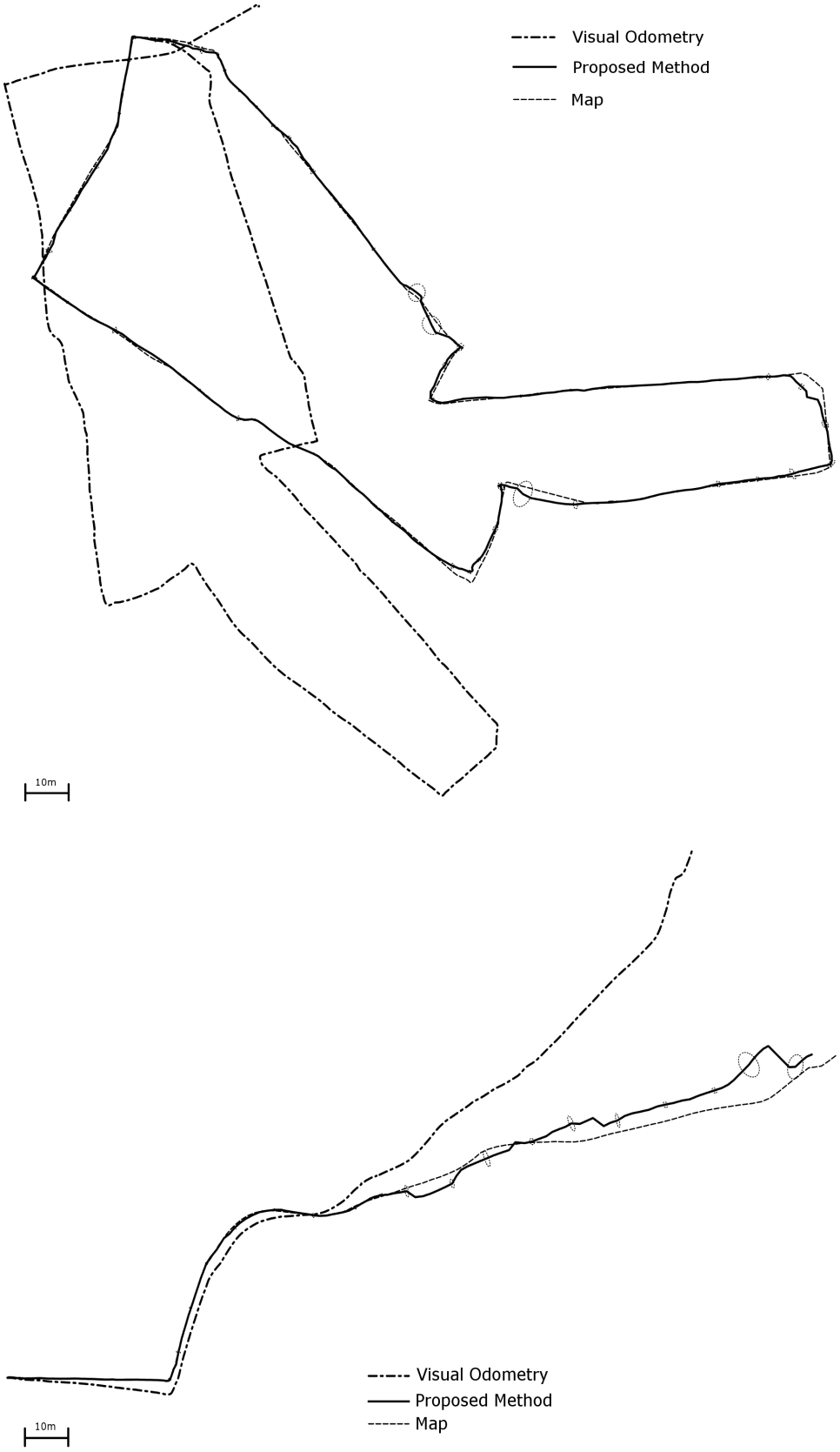


Figure 17: Localization in Crowded Urban Environments.

method will not work in environments without 3D feature nor salient line segments. This issue was seen in our experiments in a broad street. We could work around some of this issue using a camera with higher resolution and larger baseline. In the future we hope to improve our method by introducing other features or combining it with radically different approaches to solve this issue.

Another problematic condition is, as can be seen in Section 5.5, extremely adverse illuminations under which not enough edge points can be found. A straightforward approach to this issue is to extend the dynamic range of the camera. We plan to investigate whether high dynamic range cameras solve this issue.

6.1 Comparison with Existing Methods

We evaluated the performance of two existing methods; one is SIFT keypoint matching and the other is FAB-MAP. In the last decade, SIFT local feature has become popular in computer vision. SIFT is known as a feature descriptor which is robust to illumination changes compared to other image features such as like color, histogram and eigenimage which have been previously employed.

FAB-MAP, one of the state-of-the-art methods of place recognition, uses Speed Up Robust Features (SURF) which is also known as an illumination invariant feature. FAB-MAP is robust to illumination changes and partial occulusion because of bag-of-words approach and Chow-Liu trees which handle conditional dependencies between visual words. Neither SIFT itself nor FAB-MAP provide precise position of the robot, so we cannot directly compare them with our method; however, it should be interesting to know how these methods work under adverse illumination conditions which we are trying to address. We used publicly available SIFT and FAB-MAP implementations [6] [28].

We extracted 48 pairs of images of the same place from (a) dataset used in Section 5.5, to extract and match SIFT keypoints. The result is shown in Fig. 18. Only 6 from 48 image pairs (12.5%) correctly matched keypoints more than 10; pairs with no correct matches were 22 from 48 image pairs (45.9%). The result can be understood that SIFT descriptors changed significantly because of illumination changes.

The performance of FAB-MAP is evaluated in the same environment. We collected images at 11:00, 12:00 and 16:00 on a sunny day. FAB-MAP calculates probabilities that an image is coming from previously visited places or new place. If the probability exceeds threshold (0.99), the images are considered as the same place. Using 12:00 and 16:00 dataset, images recalled correctly was 4 from 1818 images (0.22%). Moreover, correct recalls are obtained only when illumination changes are relatively small (see Fig. 19). With datasets with similar illumination conditions (11:40 and 12:00), the recall rate was considerably better (10.9%). Although FAB-MAP is robust to small or partial illumination changes, its performance is largely degraded under extremely adverse illumination conditions in which image features are significantly distorted.

This time, the camera is mounted perpendicular to the direction of the movement as described in FAB-MAP literature. It should be noted that the camera we used had a small field of view and relatively

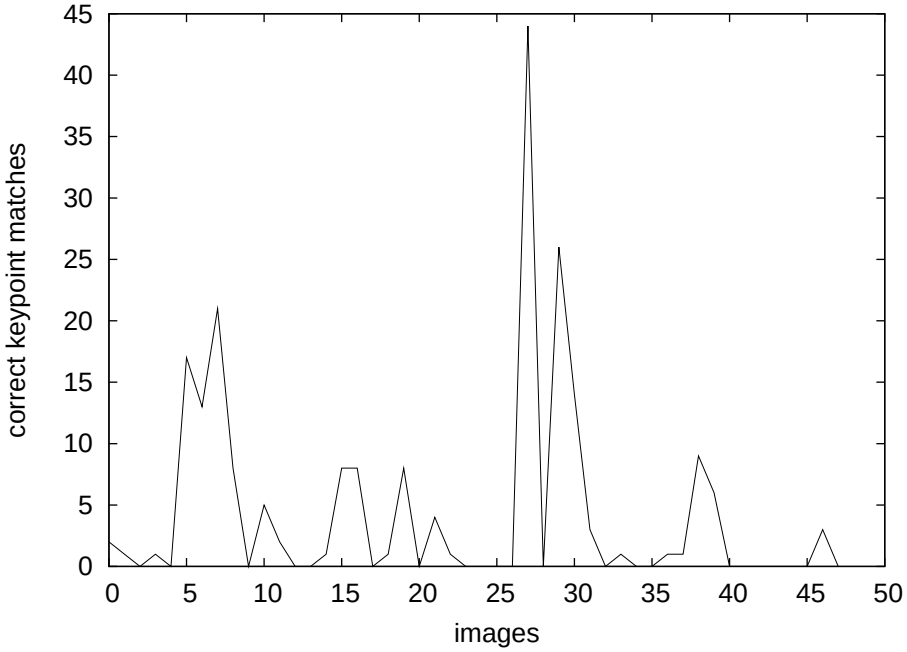


Figure 18: Result of SIFT keypoint matching

small dynamic range. A camera with larger field of view and higher dynamic range may improve the performance.

We could combine our method with FAB-MAP so that two methods compensate each other. FAB-MAP provides topological localization, not precise position of the robot, it does not require 3D shape information. In contrast, our method provides precise localization and requires 3D shape information. For the purpose of navigation in a wide open space without 3D shape, precise position of the robot may not be indispensable. Also, FAB-MAP could be useful to provide initial position estimation for our method, since our method has only position tracking feature and global localization is not implemented.

7 Conclusion

In this paper, we have proposed a new localization method for outdoor navigation using a stereo camera only. The proposed method works robustly under various illumination conditions due to map-matching using 2D grid maps generated from 3D point clouds obtained by a stereo camera. We incorporated salient line segments extracted from the ground into the grid maps, making it possible to localize in environments without 3D features. Experimental results showed the effectiveness and robustness of the proposed method under various weather and illumination conditions.

REFERENCES

[1] Martin Adams, Sen Zhang, and Lihua Xie. Particle filter based outdoor robot localization using natural features extracted from laser scanners. In *Proc. of the IEEE Int. Conf. on Robotics &*

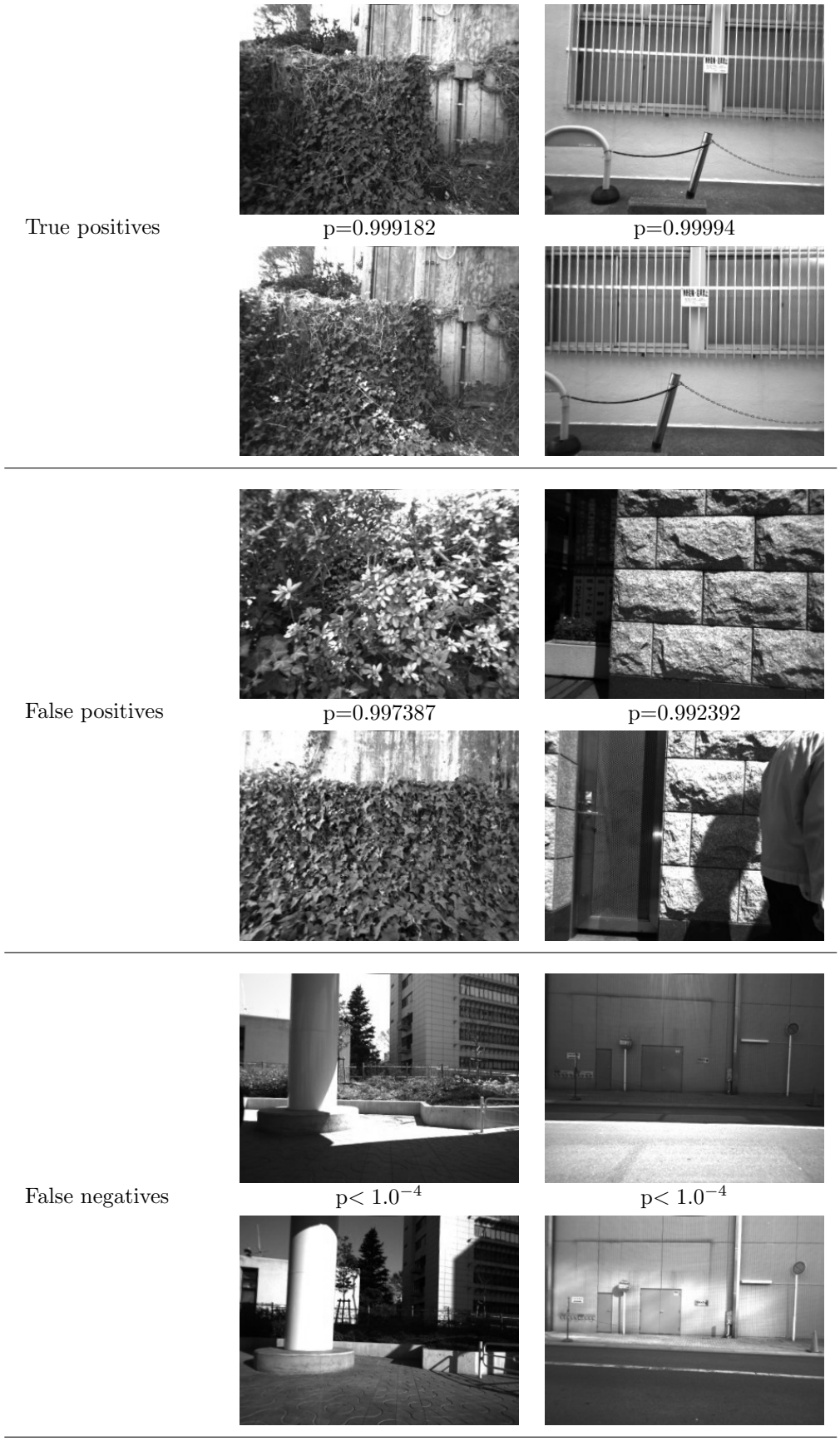


Figure 19: The result of FAB-MAP.

Automation (ICRA), pp. 1493–1498, (2004).

[2] Kazunori Ohno, Takashi Tsubouchi, Bunji Shigematsu, Shoichi Maeyama, and Shin'ichi Yuta. Outdoor navigation of a mobile robot between buildings based on dgps and odometry data fusion. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pp. 1978–1984, (2003).

[3] Jonathan R. Schoenberg, Mark Campbell, and Isaac Miller. Localization with multi-modal vision measurements in limited GPS environments using gaussian sum filters. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pp. 1423–1428, (2009).

[4] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. of Computer Vision*, 60(2):91–110, (2004).

[5] Stephen Se, David G. Lowe, and James J. Little. Vision-based global localization and mapping for mobile robots. *IEEE Trans. on Robotics*, 21:364–375, (2005).

[6] David Lowe. Demo software: SIFT keypoint detector. `http://www.cs.ubc.ca/~lowe/keypoints/`, (2005).

[7] C. Thorpe, M.H. Hebert, T. Kanade, and S.A. Shafer. Vision and navigation for the carnegie-mellon navlab. *IEEE Trans. on Pattern Analysis & Machine Intelligence*, 10(3):362 –373, (1988).

[8] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *Proc. of IEEE Computer Society Conference on Vision & Pattern Recognition*, volume 1, pp. 652 – 659, (2004).

[9] A. Chilian and H. Hirschmuller. Stereo camera based navigation of mobile robots on rough terrain. In *Proc. of the IEEE Int. Conf. on Intelligent Robots & Systems (IROS)*, pp. 4571–4576, (2009).

[10] Chunzhao Guo, S. Mita, and D. McAllester. Stereovision-based road boundary detection for intelligent vehicles in challenging scenarios. In *Proc. of the IEEE Int. Conf. on Intelligent Robots & Systems (IROS)*, pp. 1723 –1728, (2009).

[11] A. Georgiev and P.K. Allen. Vision for mobile robot localization in urban environments. In *Proc. of the IEEE Int. Conf. on Intelligent Robots & Systems (IROS)*, volume 1, pp. 472 – 477, (2002).

[12] Tomoaki Yoshida, Akihisa Ohya, and Shin'ichi Yuta. Braille block detection for autonomous mobile robot navigation. In *Proc. of the IEEE Int. Conf. on Intelligent Robots & Systems (IROS)*, pp. 633 – 638, (2000).

[13] Eric Royer, Maxime Lhuillier, Michel Dhome, and Jean-Marc Lavest. Monocular vision for mobile robot localization and autonomous navigation. *Int. J. of Computer Vision*, 74(3):237–260, (2007).

[14] M. Agrawal and K. Konolige. Real-time localization in outdoor environments using stereo vision and inexpensive gps. In *Proc. of 18th International Conference on Pattern Recognition (ICPR)*, volume 3, pp. 1063–1068, (2006).

[15] H. Katsura, J. Miura, M. Hild, and Y. Shirai. A view-based outdoor navigation using object recognition robust to changes of weather and seasons. In *Proc. of the IEEE Int. Conf. on Intelligent Robots & Systems (IROS)*, volume 3, pp. 2974 – 2979, oct. 2003.

[16] Yoichiro Yamagi, Junichi Ido, Kentaro Takemura, Yoshio Matsumoto, Jun Takamatsu, and Tsukasa Ogasawara. View-sequece based indoor/outdoor navigation robust to illumination changes. In *Proc. of the IEEE Int. Conf. on Intelligent Robots & Systems (IROS)*, pp. 1229–1234, (2009).

[17] Mark Cummins and Paul Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 6(27):647–665, (2008).

[18] Christoffer Valgren and Achim J. Lilienthal. SIFT, SURF & seasons: Appearance-based long-term localization in outdoor environments. *Robotics and Autonomous Systems*, 58(2):149–156, (2010).

[19] Masahiro Tomono. Robust 3D slam with a stereo camera based on an edge-point ICP algorithm. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pp. 4306–4311, (2009).

[20] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, (2000).

[21] J. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis & Machine Intelligence*, 8(6):679–698, (1986).

[22] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, (1997).

[23] Eijiro Takeuchi and Takashi Tsubouchi. Multi sensor map building based on sparse linear equations solver. In *Proc. of the IEEE Int. Conf. on Intelligent Robots & Systems (IROS)*, pp. 2511–2518, (2008).

[24] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, volume 2, pp. 1322 – 1328, (1999).

[25] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11:391–427, (1999).

[26] Scott Lenser and Manuela Veloso. Sensor resetting localization for poorly modelled mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, volume 2, pp. 1225 –1232, (2000).

[27] R. Ueda, T. Arai, K. Sakamoto, T. Kikuchi, and S. Kamiya. Expansion resetting for recovery from fatal error in monte carlo localization - comparison with sensor resetting methods. In *Proc. of the IEEE Int. Conf. on Intelligent Robots & Systems (IROS)*, volume 3, pp. 2481 – 2486, (2004).

[28] Mark Cummins. Fab-map binaries. `http://www.robots.ox.ac.uk/~mjc/Software.htm`.