3

Three-Dimensional Environment Modeling Based on Structure from Motion with Point and Line Features by Using Omnidirectional Camera

Ryosuke Kawanishi, Atsushi Yamashita and Toru Kaneko
Shizuoka University
Japan

1. Introduction

Three-dimensional map is available for autonomous robot navigation (path planning, self-localization and object recognition). In unknown environment, robots should measure environments and construct their maps by themselves.
Three-dimensional measurement using image data makes it possible to construct an environment map (Davison, 2003). However, many environmental images are needed if we use a conventional camera having a limited field of view (Ishiguro et al., 1992). Then, an omnidirectional camera is available for wide-ranging measurement, because it has a panoramic field of view (Fig. 1). Many researchers showed that an omnidirectional camera is effective in measurement and recognition in environment (Bunschoten & Krose, 2003; Geyer & Daniilidis, 2003; Gluckman & Nayar, 1998).

Fig. 1. Omnidirectional camera equipped with a hyperboloid mirror. The left figure shows an acquired image.

Our proposed method is based on structure from motion. Previous methods based on structure from motion often use feature points to estimate camera movement and measure environment (Rachmielowski et al., 2006; Kawanishi et al., 2009). However, many non-textured objects may exist in surrounding environments of mobile robots. It is hard to extract enough number of feature points from non-textured objects. Therefore, in an environment having non-textured objects, it is difficult to construct its map by using feature points only.

Then, line features should be utilized for environment measurement, because non-texture objects often have straight-lines. As examples of previous works using lines, a method for precise camera movement estimation by using stereo camera (Chandraker et al., 2009), a method for buildings reconstruction by using orthogonal lines (Schindler, 2006) and so on (Bartoli & Sturm, 2005; Smith et al., 2006; Mariottini & Prattichizzo, 2007) have been proposed.
However, there is a prerequisite on previous line detections of them. A method must obtain a vanishing point (Schindler, 2006) or a pair of end points of the straight-line (Smith et al., 2006). Some of previous line detection is only for a normal camera (Chandraker et al., 2009). Alternatively, some previous methods obtain line correspondences by hand (Bartoli & Sturm, 2005; Mariottini & Prattichizzo, 2007).
We propose a method for straight-line extraction and tracking on distorted omnidirectional images. The method does not require a vanishing point and end points of straight-lines. These straight-lines are regarded as infinite lines in the measurement process (Spacek, 1986). Therefore, the proposed method can measure straight-lines even if a part of the line is covered by obstacles during its tracking.
Our proposed method measures feature points together with straight-lines. If only straight-lines are used for camera movement estimation, a non-linear problem must be solved. However, camera movement can be estimated easily by a linear solution with point correspondences. Moreover, although few numbers of straight-lines may be extracted from textured objects, many feature points will be extracted from them. Therefore, we can measure the environment densely by using both feature points and straight-lines.
The process of our proposed method is mentioned below (Fig. 2). First, feature points and straight-lines are extracted and tracked along an acquired omnidirectional image sequence. Camera movement is estimated by point-based Structure from Motion. The estimated camera movement is used for an initial value for line-based measurement.

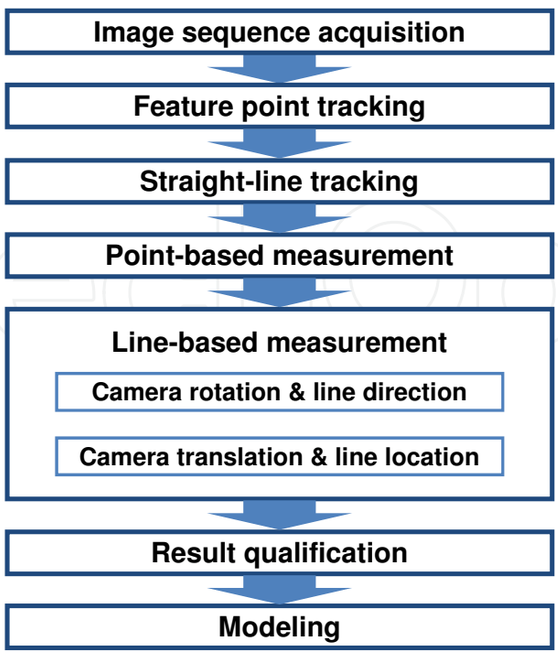


Fig. 2. Procedure of our proposed method.

The proposed line-based measurement is divided into two phases. At the first phase, camera rotation and line directions are optimized. Line correspondence makes it possible to estimate camera rotation independently of camera translation (Spacek, 1986). Camera rotation can be estimated by using 3-D line directions. At the second phase, camera translation and 3-D line location are optimized. The optimization is based on Bundle adjustment (Triggs et al., 1999). Some of measurement results have low accuracy. The proposed method rejects such results.

Measurement results of feature points and straight-lines are integrated. Triangular meshes are generated from the integrated measurement data. By texture-mapping to these meshes, a three-dimensional environment model is constructed.

2. Coordinate system of omnidirectional camera

The coordinate system of the omnidirectional camera is shown in Fig. 3. A ray heading to image coordinates (u,v) from the camera lens is reflected on a hyperboloid mirror. In this paper, the reflected vector is called a ray vector. The extension lines of all ray vectors intersect at the focal point of the hyperboloid mirror. The ray vector $\mathbf{r}$ is calculated by the following equations.

$$\mathbf{r} = \begin{bmatrix} \lambda(u - c_x)p_x \\ \lambda(v - c_y)p_y \\ \lambda f - 2\gamma \end{bmatrix} \tag{1}$$

$$\lambda = \frac{\alpha^2\left(f\gamma + \beta\sqrt{u^2 + v^2 + f^2}\right)}{\alpha^2 f^2 - \beta^2\left(u^2 + v^2\right)} \tag{2}$$

where c_x and c_y are the center coordinates of the omnidirectional image, p_x and p_y are pixel size, f is the focal length, α, β and γ are hyperboloid parameters. These parameters are calibrated in advance.

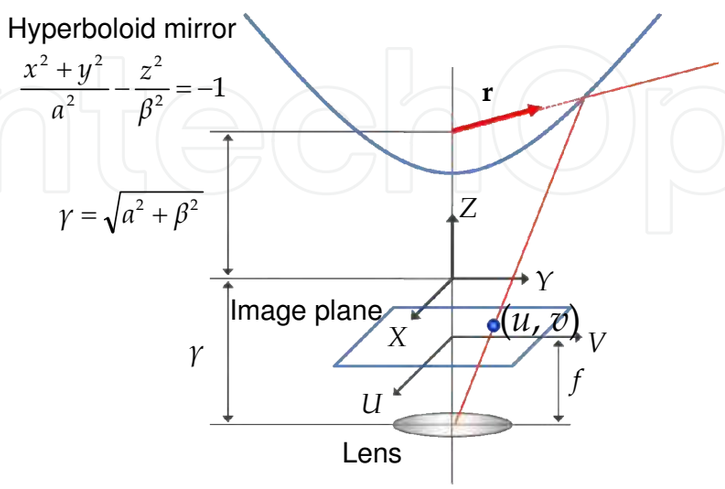
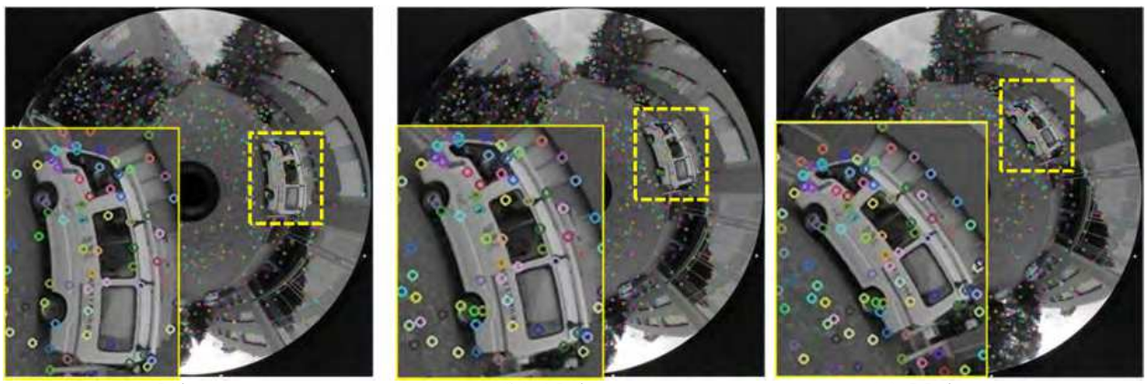


Fig. 3. The coordinate system of the omnidirectional camera. Ray vector $\mathbf{r}$ is defined as a unit vector which starts from the focal point of a hyperboloid mirror.

3. Feature tracking

3.1 Point tracking

Feature points are tracked along an omnidirectional image sequence by KLT tracker (Shi & Tomasi, 1994). These points are used for initial estimation of camera movement and measurement for textured objects. An example of feature point tracking is shown in Fig. 4.

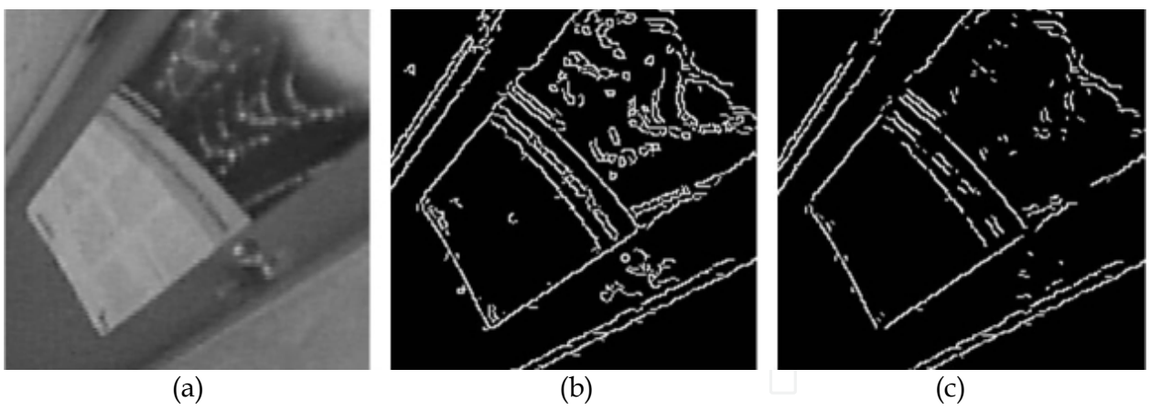


(a) 0 frame. (b) 10 frame. (c) 20 frame.

Fig. 4. An example of feature point tracking. Points which have the same color show corresponding points.

3.2 Straight-line tracking

Straight-lines are extracted from a distorted omnidirectional image. The proposed method obtains edge points by Canny edge detector (Canny, 1986). An example of edge point detection is shown in Fig. 5 (a) and (b).



(a) (b) (c)

Fig. 5. Edge segment extraction. (a) Input image. (b) Detected canny edge points. (c) Edge segments are separated by rejecting corner points.

To separate each straight-line, corner points are rejected as shown in Fig. 5 (c). Corner points are detected by using two eigenvalues of the Hessian of the image. Hessian matrix is calculated by the following equation.

$$\mathbf{H} = \begin{bmatrix} I_x{}^2 & I_x I_y \\ I_x I_y & I_y{}^2 \end{bmatrix} \tag{3}$$

where I_x and I_y are derivatives of image I. An edge point which has large value of the ratio of eigenvalues is regarded as a point locating on a line. In the proposed method, if the ratio is smaller than 10, the edge point is rejected as a corner point. This process provides us separated edge segments.

A least square plane is calculated from ray vectors of edge points which belong to an edge segment. If the edge segment constitutes a straight-line, these ray vectors are located on a plane (Fig. 6). Therefore, an edge segment which has a small least square error is regarded as a straight-line. The proposed method is able to extract straight-lines, even if an edge segment looks like a curved line in a distorted omnidirectional image.

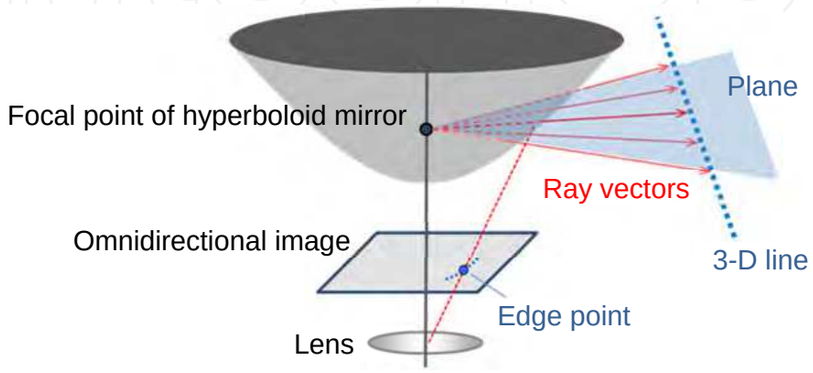


Fig. 6. The relationship between a straight-line and a ray vector.

The maximum number of edge points which satisfy equation (4) is calculated by using RANSAC (Fischler & Bolles, 1981).

$$\left(\mathbf{r}_{i,j}{}^{\mathrm{T}}\mathbf{n}_i\right)^2 < l_{\mathrm{th}} \tag{4}$$

where l_{th} is a threshold. $\mathbf{r}_{i,j}$ is a ray vector heading to an edge point j included in an edge segment i. $\mathbf{n}_i$ is the normal vector of the least square plane calculated from the edge segment i. If over half the edge points of the edge segment i satisfy equation (4), the edge segment is determined as a straight-line. The threshold l_{th} is calculated by the following equation.

$$l_{\mathrm{th}} = \cos^2\left(\frac{2\pi}{2r_{\mathrm{m}}\pi}\right) = \cos^2\left(\frac{1}{r_{\mathrm{m}}}\right) \tag{5}$$

where r_{m} is the radius of projected mirror circumference in an omnidirectional image. A threshold l_{th} allows angle error within $1/r_{\mathrm{m}}$ [rad]. It means that an angle error between a ray vector and a straight-line is within 1 pixel.

Straight-lines are tracked along an omnidirectional image sequence. The proposed method extracts points at constant intervals on a straight-line detected in the current frame (Fig. 7 (a) and (b)). These points are tracked to the next frame by KLT tracker (Fig. 7 (d)). Edge segments are detected in the next frame (Fig. 7 (c)). The edge point closest to the tracked point is selected as a corresponding edge point (Fig. 7 (e)). The edge segment which has themaximum number of corresponding edge points is regarded as a corresponding edge segment (Fig. 7 (f)). If an edge segment corresponds to several lines, a line which has larger number of corresponding edge points is selected.

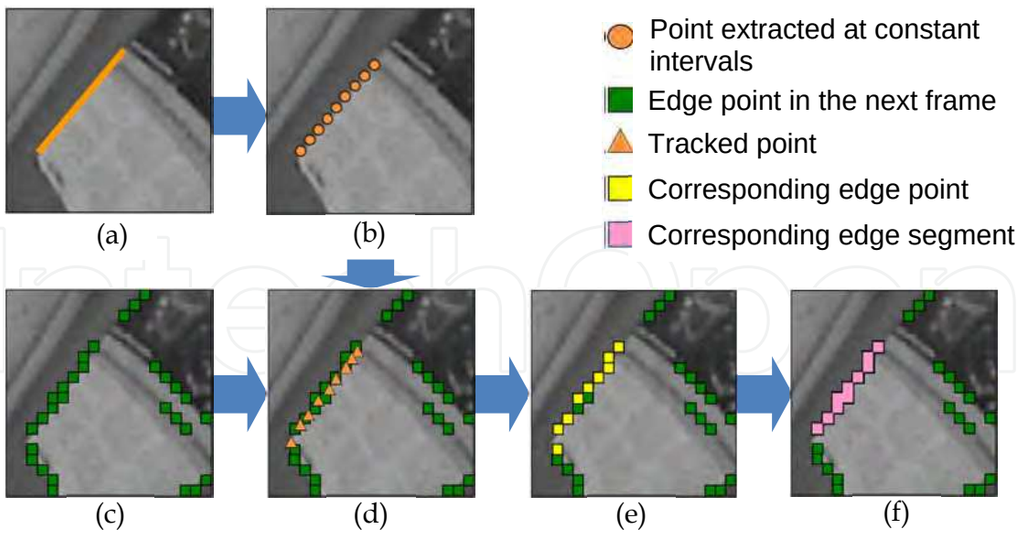
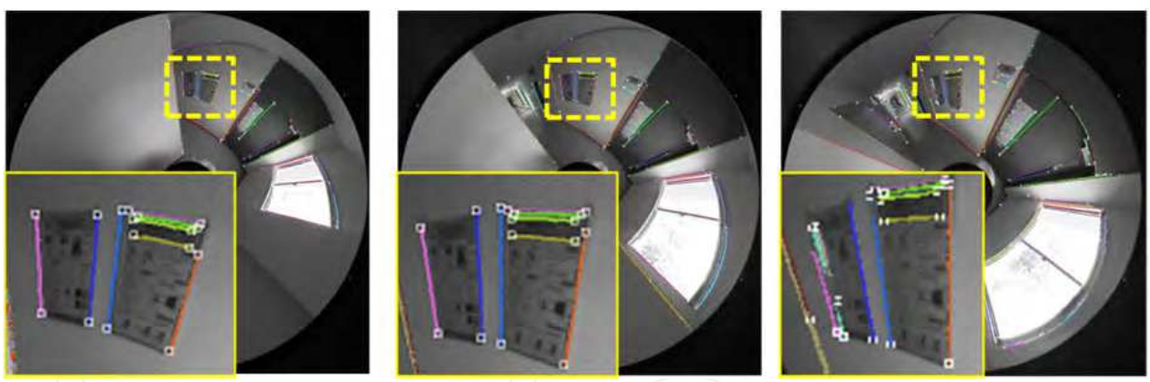


Fig. 7. Searching for a corresponding edge segment in the next frame. (a) Straight-line extracted in the current frame. (b) Points extracted at constant intervals on the line. (b) Edge segments in the next frame. (c) Points (b) are tracked between the current frame and the next frame. (d) Corresponding edge points. (e) Corresponding edge segment.



(a) 0 frame. (b) 10 frame. (c) 20 frame.

Fig. 8. An example of straight-line tracking. Lines which have a same color show corresponding lines. Although an end point of a line is shown as a white square, it is not used for straight-line detection.

Matching point search on a line has the aperture problem (Nakayama, 1988). However, it is not difficult for the proposed method to obtain corresponding lines, because it does not require point-to-point matching. By continuing the above processes, straight-lines are tracked along the omnidirectional image sequence. An example of line tracking is shown in Fig. 8.

4. Environment measurement

4.1 Point-based measurement

Camera movement is estimated by a point-based method (Kawanishi et al., 2009). The method is based on eight-point algorithm (Hartley, 1997).

An essential matrix $\mathbf{E}$ is calculated from ray vectors of corresponding feature points. An essential matrix $\mathbf{E}$ and ray vectors satisfy the following equation.

$$\mathbf{r}_i \mathbf{E} \mathbf{r}'_i = 0 \tag{6}$$

where ray vectors $\mathbf{r}_i = [x_i, y_i, z_i]^{\mathrm{T}}$ and $\mathbf{r}'_i = [x'_i, y'_i, z'_i]^{\mathrm{T}}$ are those of the corresponding point in two images. Camera rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ are calculated from essential matrix $\mathbf{E}$ by singular value decomposition. Equation (6) is transformed as follows.

$$\mathbf{u}^{\mathrm{T}} \mathbf{e} = 0 \tag{7}$$

where

$$\mathbf{u} = [x_i x'_i, y_i x'_i, z_i x'_i, x_i y'_i, y_i y'_i, z_i y'_i, x_i z'_i, y_i z'_i, z_i z'_i]^{\mathrm{T}},$$

$$\mathbf{e} = [e_{11}, e_{12}, e_{13}, e_{21}, e_{22}, e_{23}, e_{31}, e_{32}, e_{33}]^{\mathrm{T}}.$$

e_{ab} is the row a and column b element of Essential matrix $\mathbf{E}$. The matrix $\mathbf{E}$ is obtained by solving simultaneous equations for more than eight pairs of corresponding ray vectors.

$$J = \|\mathbf{U}\mathbf{e}\|^2 \rightarrow \min \tag{8}$$

where $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_n]^{\mathrm{T}}$. n is the number of feature points. $\mathbf{e}$ is calculated as the eigenvector of the smallest eigenvalues of $\mathbf{U}^{\mathrm{T}}\mathbf{U}$. Estimated camera movement in this process is used as an initial value for line-based measurement. However, not all feature points tracked in the image sequence correspond satisfactorily due to image noise, etc. Mistracked feature points should be rejected. The proposed method rejects these points as outliers by using RANSAC algorithm (Fischler & Bolles, 1981).

4.2 Line-based measurement

Estimated camera movement is optimized by using straight-lines. A straight-line is represented as infinite lines by using its direction vector $\mathbf{d}^w$ and location vector $\mathbf{l}^w$ ($\mathbf{l}^w + k\mathbf{d}^w$, k is a factor). The superscript w means that the vector is in world coordinate system. As a prerequisite for line-based measurement, at least, more than 3 images and 3 pairs of corresponding lines (at least one line is not parallel to others) are needed. In the first step, camera rotation and line directions are estimated. The step is independent of camera translation and line locations estimation. In the next step, camera translation and line locations are optimized by a method based on Bundle adjustment (Triggs et al., 1999). In these phases, initial value of 3-D line direction and location are required. These initial values are calculated from line correspondences and initial camera movements.

4.2.1 Camera rotation and 3-D line direction optimization

Our proposed method calculates a normal vector $\mathbf{n}_i^c$ of a least square plane calculated from an edge segment i in Section 3.2. The superscript c means that the vector is in a camera coordinate system at camera position c. Camera rotation depends on 3-D line direction vector $\mathbf{d}_i^w$ and normal vector $\mathbf{n}_i^c$. By using initial values of camera rotation and normal

vectors $\mathbf{n}_i^c$, a sum of errors $E_{\mathbf{R}}$ between camera rotation matrix $\mathbf{R}_c^w$ and 3-D line direction $\mathbf{d}_i^w$ are calculated as shown in the following equation.

$$E_{\mathbf{R}} = \sum_c \sum_i \left(\left(\mathbf{R}_c^{w\mathrm{T}} \mathbf{n}_i^c \right)^{\mathrm{T}} \mathbf{d}_i^w \right)^2 \tag{9}$$

where, $\mathbf{R}_c^w$ is a rotation matrix from the world coordinate system to camera coordinate system c. Here, $\mathbf{d}_i^w$ and $\mathbf{n}_i^c$ are unit vectors. The relationship between a direction vector and a normal vector is shown in Fig. 9. Camera rotations and line directions are optimized by minimizing $E_{\mathbf{R}}$. Levenburg-Marquardt method is used for the minimization.

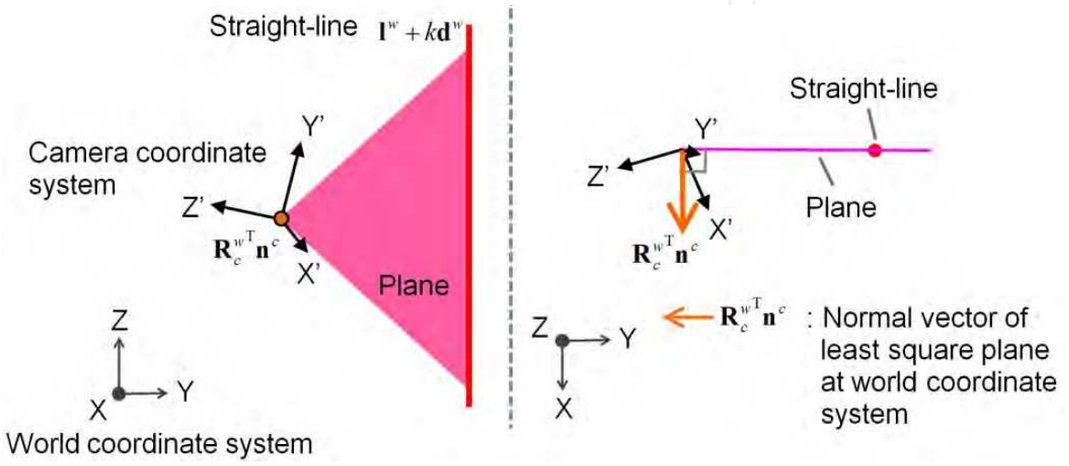


Fig. 9. Relationship between a direction vector of straight-line and a normal vector of a least square plane.

4.2.2 Camera translation and 3-D line location optimization

Camera translation vector $\mathbf{t}_c^w$ and 3-D line location $\mathbf{l}_i^w$ are optimized by Bundle adjustment (Triggs et al., 1999). The method estimates camera movements by minimizing reprojection errors. The projection error of the straight-line is calculated as an angle error between two vectors on a plane which is orthogonal to the line direction. The sum of reprojection errors of straight-lines $E_{\mathbf{t}}$ is calculated by the following equation.

$$E_{\mathbf{t}} = \sum_c \sum_i \left(1 - \hat{\mathbf{l}}_i^{c\mathrm{T}} \mathbf{g}_i^c \right)^2 \tag{10}$$

where $\mathbf{g}_i^c$ is a vector located on the plane $\mathbf{n}_i^c$, and it crosses the 3-D line at a right angle. Thus, $\mathbf{g}_i^c$ satisfies $\mathbf{R}_c^{w\mathrm{T}} \mathbf{g}_i^c \perp \mathbf{d}_i^w$ and $\mathbf{g}_i^c \perp \mathbf{n}_i^c$. $\hat{\mathbf{l}}_i^c$ is a vector which connects the camera position c and the 3-D line location with the shortest distance. $\hat{\mathbf{l}}_i^c$ is calculated by the following equation.

$$\hat{\mathbf{l}}_i^c = \mathbf{R}_c^w \left(B_{i,c} \mathbf{d}_i^w + \mathbf{l}_i^w - \mathbf{t}_c^w \right) \Big/ \left\| B_{i,c} \mathbf{d}_i^w + \mathbf{l}_i^w - \mathbf{t}_c^w \right\| \tag{11}$$

where $B_{i,c}$ is a factor which shows a location on the 3-D line. $B_{i,c}$ satisfies the following equation.

$$\left\| \left(A_{i,c} \mathbf{R}_c^{w\mathrm{T}} \mathbf{g}_i^c + \mathbf{t}_c^w \right) - \left(B_{i,c} \mathbf{d}_i^w + \mathbf{l}_i^w \right) \right\| \rightarrow \min \tag{12}$$

The relationship between these vectors is shown in Fig. 10.

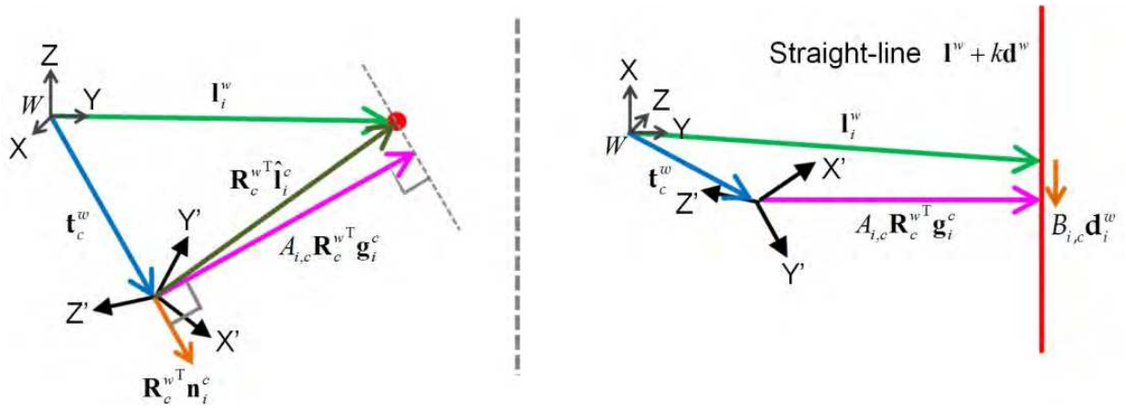


Fig. 10. Relationship between camera translation vector and 3-D line location vector.

The sum of reprojection errors of straight-lines $E_{\mathbf{t}}$ is minimized by a convergent calculation based on Levenburg-Marquardt method. In these two optimization steps, lines which have large error are rejected as outliers by RANSAC algorithm.

In the proposed method, 3-D lines are represented as uniformly-spaced points $\mathbf{e}_{i,n}^w$. 3-D coordinates of these points are calculated by the following equation.

$$\mathbf{e}_{i,n}^w = hn\mathbf{d}_i^w + \mathbf{l}_i^w \tag{13}$$

where h is a uniform distance and n is an integer number, respectively. 3-D coordinates of $\mathbf{e}_{i,n}^w$ is reprojected to the image sequence. When the 2-D coordinates of the reprojection point are close to the corresponding edge segment enough, the point is added into measurement data.

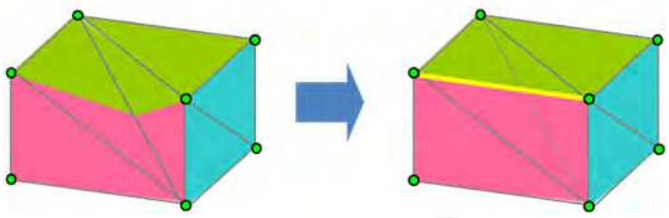
By using estimated camera movement, 3-D coordinates of feature points which have the minimal reprojection error are calculated and integrated with straight-line measurement data.

4.3 Result qualification

Measurement data which have low accuracy should be rejected before 3-D model construction. Measurement accuracy of the feature point is evaluated by following equations.

$$a_{i,m} = \left\| \frac{\partial \mathbf{p}_{i,m}}{\partial u_{c1,i}} \right\| + \left\| \frac{\partial \mathbf{p}_{i,m}}{\partial v_{c1,i}} \right\| + \left\| \frac{\partial \mathbf{p}_{i,m}}{\partial u_{c2,i}} \right\| + \left\| \frac{\partial \mathbf{p}_{i,m}}{\partial v_{c2,i}} \right\| \tag{14}$$

where $\mathbf{p}_{i,m}$ is 3-D coordinates calculated from corresponding feature points between camera position c_1 and c_2. $\left(u_{c1,i}, v_{c1,i}\right)$ and $\left(u_{c2,i}, v_{c2,i}\right)$ are image coordinates of feature points. The method calculates $a_{i,m}$ of all camera position combination. If the smallest value $a_{\min,i}$ is larger than a given threshold a_{th}, the feature point is rejected as a measurement result which has low accuracy.



(a) (b)

Fig. 11. Triangular mesh generation and its optimization. (a) Triangular meshes generated by Delaunay triangulation. (b) Optimized triangular meshes.

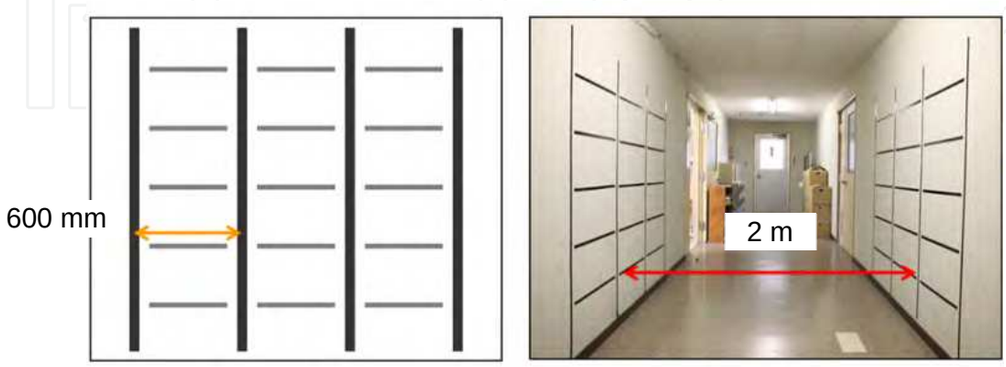
Measurement accuracy of straight-line is evaluated by equation (14), too. In this evaluation, $\mathbf{p}_{i,m}$ is the middle point of the line connecting two vectors $\mathbf{g}_i^{c1}$ and $\mathbf{g}_i^{c2}$ at the shortest distance. Image coordinates $\left(u_{c1,i}, v_{c1,i}\right)$ and $\left(u_{c2,i}, v_{c2,i}\right)$ are reprojection points of these vectors to images acquired at camera position c_1 and c_2.

5. Model construction

Triangular meshes are generated from integrated measurement data by using the 3-D Delaunay triangulation. However, Delaunay triangulation generates a triangular mesh which contradicts a physical shape because the triangular mesh does not consider the shape of the measurement object. Therefore, we apply the triangular optimization method (Nakatsuji et al., 2005) to the triangular mesh (Fig. 11). The method adapts the triangular mesh to the physical shape by detecting a texture distortion. By texture mapping to these meshes, a 3D environment model is constructed.

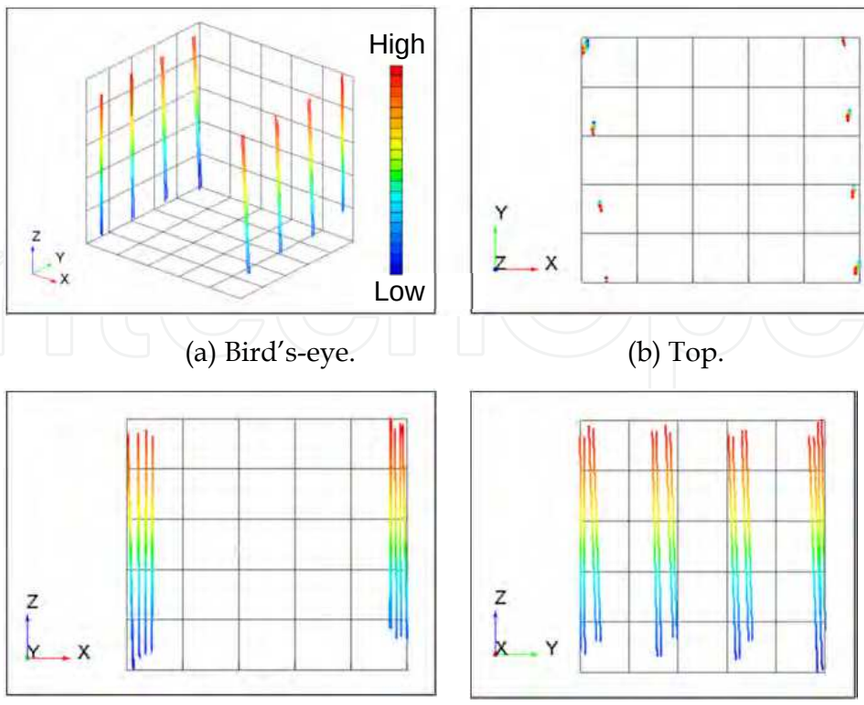
6. Experiments

First, accuracy of line-based measurement is evaluated. Measurement objects are lengthwise-lines on a flat wall shown in Fig. 12. The reason for including crosswise-lines is that the proposed method needs lines having different direction. The moving distance of the camera was about 2m. The number of input images is 72. An input image size is 2496×1664 pixels.



(a) Object. (b) Environment.

Fig. 12. Measurement objects place on flat walls. Vertical lines are measured for accuracy evaluation. Level lines are set for camera movement estimation.



(a) Bird's-eye. (b) Top.

(c) Front. (d) Side.

Fig. 13. Measurement result for accuracy evaluation. Vertical lines are measured precisely. Measurement results of level lines are removed as low measurement accuracy data.

	Angle (direction) [deg]	Angle (plane) [deg]	Depth [mm]
Standard deviation	1.2	-	2.3
Maximum	1.9	1.5	7.5

Table 1. Evaluation result of vertical line measurements.

A measurement result is shown in Fig. 13. A measurement result which has larger Z-coordinate value is displayed in red, and smaller one is displayed in blue. Angles and depth errors were calculated for evaluation of measurement accuracy in Table 1. An angle error of calculated line directions is 1.2 degree standard deviation. Its maximum error is 1.9 degree. An angle error between two flat walls estimated from measurement data is within 1.5 degrees. A depth error between an estimated flat wall and reconstructed lines has 2.3 mm standard deviation. Its maximum error is 7.5 mm. This experiment shows that our proposed method has sufficient accuracy to accomplish static obstacle avoidance, self-localization.

Next, we experimented in an environment including non-textured objects as shown in Fig. 14. We used 84 omnidirectional images. Measurement results of feature points and straight-lines are shown in Fig. 15. The blue marks in the figure show the camera trajectory. Although feature point measurement results are sparse, straight-lines can be measured densely. This experimental result shows that our proposed method is effective for a non-textured environment.

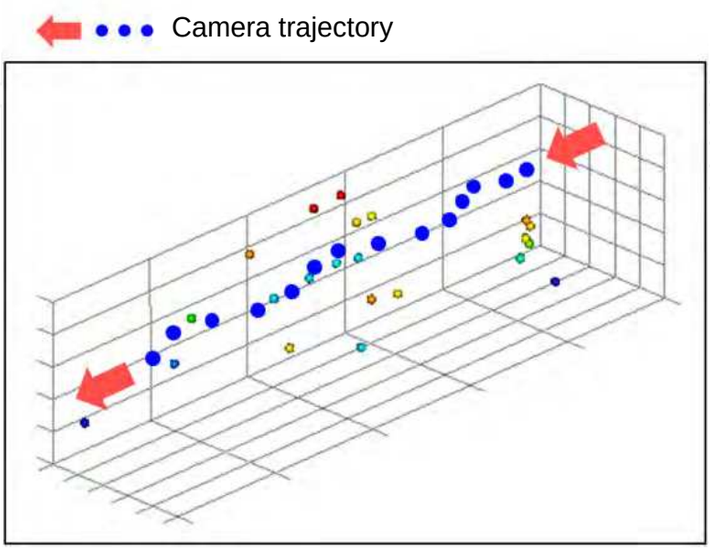
(a) Non-textured environment.

(b) Input image.

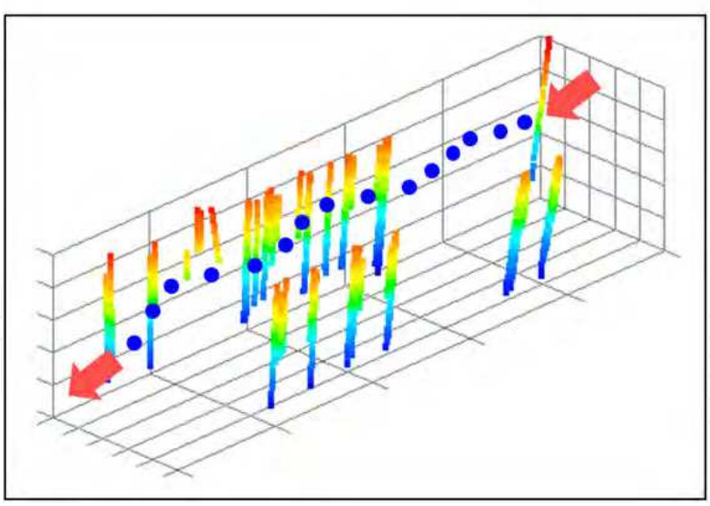
Fig. 14. Non-textured environment. We cannot get enough feature points in the environment because there are few features.

Modeling result is shown in Fig. 16. Images having a view-point which is different from camera observation points can be acquired. A model constructed from feature point measurement data is only a small part of this environment (Fig. 16(a) and (c)). Meanwhile, edge measurement data makes it possible to construct a non-textured environment model (Fig. 16}(b) and (d)).

We also experimented in an outdoor environment including textured objects (trees and so on) and non-textured objects (buildings and so on) as shown in Fig. 17. We used 240 omnidirectional images. An integrated measurement result is shown in Fig. 18. As one of textured objects, the shape of ground surface is measured by point-based measurement. As non-textured objects, the shape of the building is measured by line-based measurement.

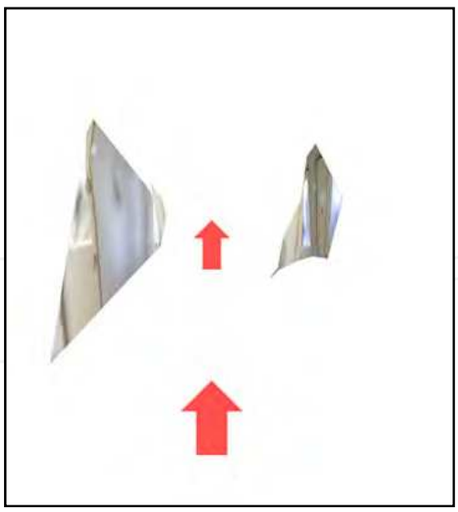


(a) Feature points.



(b) Straight-lines.

Fig. 15. Measurement results of non-textured environment. (a) Although camera movement estimation is possible, we get sparse measurement results. (b) Straight-lines make it possible to measure non-textured environments densely.



(a) Front (feature point only).

(b) Front (with straight-line).

(c) Bird's-eye (feature point only).

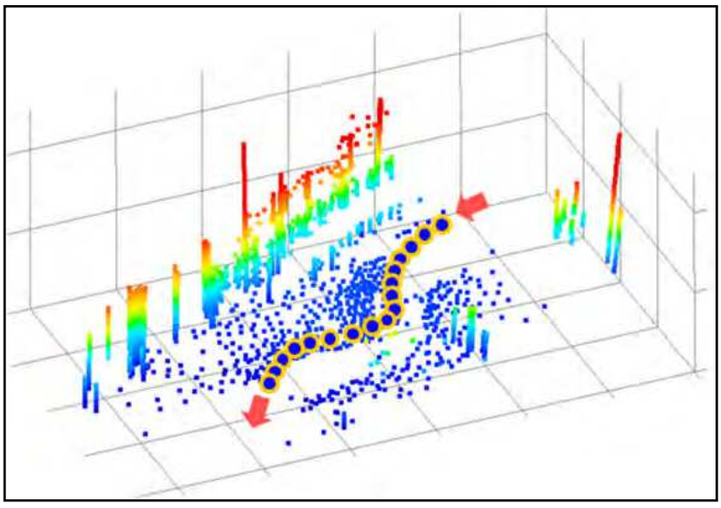
(d) Bird's-eye (with straight-line)

Fig. 16. Modeling results of non-textured environment. We can construction a model including many non-textured objects by the method with straight-lines.

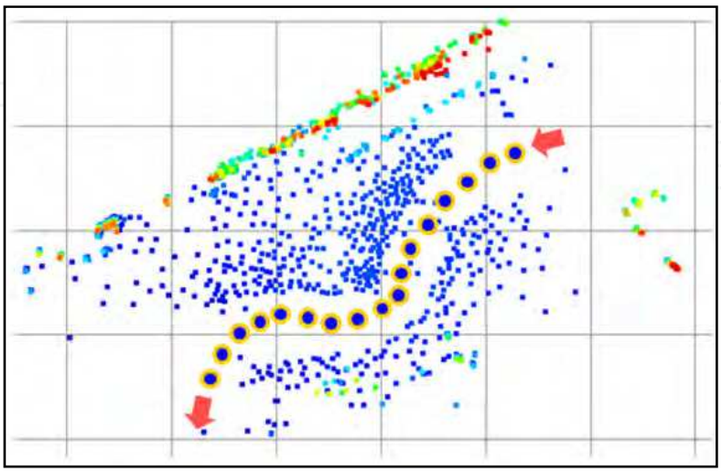
Modeling result is shown in Fig. 19. By the combination of point-based measurement and line-based measurement, our proposed method can construct a model of 3-D environment including textured and non-textured objects. Experimental results showed the effectiveness of our proposed method.

(a) Environment. (b) Input image.

Fig. 17. Outdoor environment. There are textured objects (trees, tiles and so on) and non-textured objects (walls etc.)



(a) Bird's-eye



(b) Top

Fig. 18. Measurement result of outdoor environment.

(a) Bird's-eye.

(b) Right.

(c) Front.

Fig. 19. Modeling results of outdoor environment.

7. Conclusions

We proposed an environment modeling method based on structure from motion using both feature points and straight-lines by using an omnidirectional camera. Experimental results showed that our proposed method is effective in environment including both textured and non-textured objects.

As future works, the precision improvement of edge tracking is necessary. Moreover, we should evaluate difference of camera movement estimation accuracy between point-based measurement and edge-based measurement. Further, edge position correlation should be used for increasing measurement stability.

8. References

Davison, A. J. (2003). Real-Time Simultaneous Localisation and Mapping with a Single Camera, *Proceedings of the 9th IEEE International Conference on Computer Vision*, Vol. 2, pp. 1403-1410, 2003.

Ishiguro, H. & Yamamoto, M. & Tsuji, S. (1992). Omni-Directional Stereo, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, pp. 257-262, 1992.

Gluckman, J. & Nayar, S. K. (1998). Ego-motion and Omnidirectional Cameras, *Proceedings of the 6th International Conference on Computer Vision*, pp. 999-1005, 1998.

Bunschoten, R. & Krose, B. (2003). Robust Scene Reconstruction from an Omnidirectional Vision System, *IEEE Transactions on Robotics and Automation*, Vol. 19, No. 2, pp. 351-357, 2003.

Geyer, C. & Daniilidis, K. (2003). Omnidirectional Video, *The Visual Computer*, Vol. 19, No. 6, pp. 405-416, 2003.

Rachmielowski, A. & Cobzas, D. & Jagersand, M. (2006). Robust SSD Tracking with Incremental 3D Structure Estimation, *Proceedings of the 3rd Canadian Conference on Computer and Robot Vision*, pp. 1-8, 2006.

Kawanishi, R.; Yamashita, A. & Kaneko, T. (2009). Three-Dimensional Environment Model Construction from an Omnidirectional Image Sequence, *Journal of Robotics and Mechatronics*, Vol. 21, No. 5, pp. 574-582, 2009.

Chandraker, M. & Lim, J. & Kriegman, D. (2009). Moving in Stereo: Efficient Structure and Motion Using Lines, *Proceedings of the 12th IEEE International Conference on Computer Vision*, pp. 1741-1748, 2009.

Schindler, G.; Krishnamurthy, P. & Dellaert, F. (2006). Line-Based Structure from Motion for Urban Environments, *Proceedings of the 3rd International Symposium on 3D Data Processing, Visualization, and Transmission*, pp. 846-853, 2006.

Bartoli, A. & Sturm, P. (2005). Structure-from-motion using lines: representation, triangulation, and bundle adjustment, *Computer Vision and Image Understanding*, Vol. 100, Issue 3, pp. 416-441, 2005.

Smith, P. & Reid, I. & Davison, A. (2006). Real Time Monocular SLAM with Straight lines, *Proceedings of the 17th British Machine Vision Conference*, pp. 17-26, 2006.

Mariottini, G. L. & Prattichizzo, D. (2007). Uncalibrated video compass for mobile robots from paracatadioptric line images, *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 226-231, 2007.

Spacek, L. A. (1986). Edge Detection and Motion Detection, *Image and Vision Computing*, Vol. 4, Issue 1, pp. 43-56, 1986.

Triggs, B. & McLauchlan, P. & Hartley, R. & Fitzgibbon, A. (1999). Bundle Adjustment -A Modern Synthesis, *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, Springer-Verlag LNCS 1883, pp. 298-372, 1999.

Shi, J. & Tomasi, C. (1994). Good Features to Track, *Proceedings of the 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 593-600, 1994.

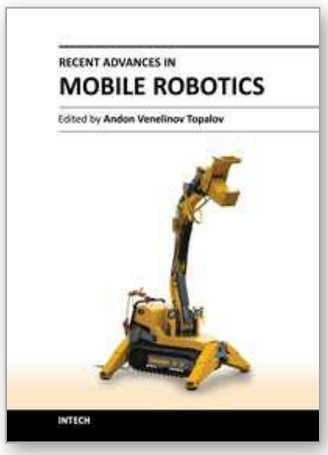
Canny, J. F. (1986). A Computational Approach to Edge Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 6, pp. 679-698, 1986.

Fischler, M. A. & Bolles, R. C. (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, *Communications of the ACM*, Vol. 24, No. 6, pp. 381-395, 1981.

Nakayama, K. & Silverman, G. (1988). The Aperture Problem -- II. Spatial Integration of Velocity Information Along Contours, *Vision Research*, Vol. 28, No. 6, pp. 747-753, 1988.

Hartley, R. (1997). In defence of the eight-point algorithm, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 6, pp. 580-593, 1997.

Nakatsuji, A. & Sugaya, Y. & Kanatani, K. (2005), Optimizing a Triangular Mesh for Shape Reconstruction from Images, *IEICE Transactions on Information and Systems*, Vol. E88-D, No. 10, pp. 2269-2276, 2005.



Recent Advances in Mobile Robotics

Edited by Dr. Andon Topalov

ISBN 978-953-307-909-7
Hard cover, 452 pages
Publisher InTech
Published online 14, December, 2011
Published in print edition December, 2011

Mobile robots are the focus of a great deal of current research in robotics. Mobile robotics is a young, multidisciplinary field involving knowledge from many areas, including electrical, electronic and mechanical engineering, computer, cognitive and social sciences. Being engaged in the design of automated systems, it lies at the intersection of artificial intelligence, computational vision, and robotics. Thanks to the numerous researchers sharing their goals, visions and results within the community, mobile robotics is becoming a very rich and stimulating area. The book Recent Advances in Mobile Robotics addresses the topic by integrating contributions from many researchers around the globe. It emphasizes the computational methods of programming mobile robots, rather than the methods of constructing the hardware. Its content reflects different complementary aspects of theory and practice, which have recently taken place. We believe that it will serve as a valuable handbook to those who work in research and development of mobile robots.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen