Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Mobile Robot Self-Localization and Local Map Alignment with a Dempster Shafer Sensor Fusion Algorithm

Hyunki Lee, Xingyong Song and Hyungsuck Cho

** Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Korea Tel: +82-42-869-3253; E-mail: {(hklee,songxy, hscho)@lca.kaist.ac.kr}*

Abstract: An important and challenging research issue associated with mobile robots is simultaneous localization and map building (SLAM), which refers to the mobile robot's capability of estimating its poses in the environment without external information, and simultaneous alignment of the local maps. To solve these kinds of research problems, researchers have proposed various methods such as odometry measurement, landmark matching, laser range image matching and a scale-invariant feature transform (SIFT)–based algorithm, all of which suffer from inevitable drawbacks such as a local minimum problem and a lack of SIFT features. Our solution for these problems is a sensor fusion method that uses a Dempster Shafer algorithm to fuse both the laser range information and the SIFT features information for the SLAM. Through a series of experiments, we tested and evaluated the proposed method. By real experiments, we analyzed the parameters of the ICP and SIFT features and we checked the robustness of our algorithm.

1. INTRODUCTION

Knowing the position and orientation in the operating environment is an essential capability for any robot that exhibits coherent, goal-oriented behavior. The basic principle of self-localization is based on a comparison of perceptual information derived from sensor readings with a priori or previously acquired knowledge about the environment. The self-localization problem has been studied intensively in the mobile robot community, and a wide variety of approaches with distinct methods and capabilities has been developed. (Dixon et al. 1997).

Some researchers have generated the idea of detecting and matching landmarks (either natural or artificial) in the environment. When natural landmarks are used, the main problem is the difficulty of detecting and matching characteristic features from sensory inputs. The sensor of choice for this task is computer vision. Most computer vision–based natural landmarks are long vertical edges, such as doors and wall junctions. However, in some cases, the natural landmark features are not easy to detect, especially in unstructured environments. When artificial landmarks are used, it is much easier to detect the landmarks because the landmarks are designed for optimal contrast. However, this method needs human interference.

To solve these problems, some research groups have proposed the idea of matching two range image maps obtained separately from two consecutive poses of a mobile robot. (Surmann et al. 2003; Downs et al. 2003) The main task is to align the overlapping parts of the range images together and subsequently achieve the translation and rotation matrix between the two consecutive poses of the mobile robot. The most popular matching or registering algorithm is called the iterative closest point (ICP) algorithm. (Besl et al. 1992)

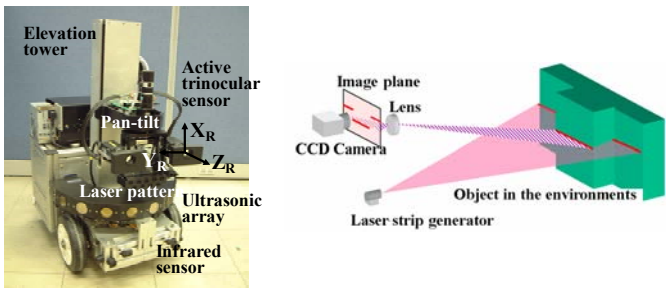
The alignment of two range images obtained from two poses faces two main problems. First, only parts of the two range images can be overlapped together. Thus, any effort to align these outlier points is a waste of time and disturbs the registration process. Second, the ICP algorithm is not robust to noise and outliers. Note in particular that because our mobile robot uses a laser structured light laser range finder, the maximum range of which is 1.7 m, the matching by the ICP method is more difficult. We therefore propose a trimmed iterative closest point (TrICP) algorithm, which can help solve the outlier point problem in the matching process.

Some researchers have recently used scale-invariant features to implement mobile robot localization and mapping. In their approach, they first extract scale-invariant feature transform (SIFT) features and then match the SIFT features. This algorithm is robust when many SIFT features exist in the environment but not if insufficient features can be detected. In addition, the map built with this method is not usable for obstacle avoidance and next view generation.

To solve the problems of the ICP-based simultaneous localization and map building (SLAM) and the SIFT-based SLAM, we propose a fusion algorithm that combines these two methods. This paper is organized as follows. In Section 2, we introduce the mobile robot and its sensor system. In Section 3, we briefly introduce the basic concept and main procedures of the TrICP algorithm. In Section 4, we discuss the use of using the SIFT method in robot localization. The fusion algorithm is discussed in Section 5. Finally, Section 6 offers a summary of the proposed algorithm along with the experimental results.

2. THE SENSOR SYSTEM OF THE MOBILE ROBOT

As shown in Fig. 1(a), our mobile robot mainly consists of an on-board PC; 2 CCD cameras, which are used for stereo vision; 2 band-pass filters; an ultrasonic sensor; and several laser structured light projectors.



(a) mobile robot (b) laser range finder

Fig. 1. Mobile robot LCAR III and the configuration of the laser range finder

The sensing system of the mobile robot has two parts; 1) a laser structured light range finder, which consists of a CCD camera, a horizontal laser light, and a band-pass filter. 2) a stereo vision sensor that consists of 2 CCD cameras. (Cho 2005)

3. SLAM ALGORITHM WITH A TRICP ALGORITHM

3.1 Laser structured light range finder

In previous chapter, we referred to the laser structured light sensor shown in Fig. 1(b). (Izquierdo et al. 1999) As shown in Fig. 1(b), the laser structured light range finder is mainly composed of a horizontal laser and a CCD camera. When the horizontal laser line is projected onto the object, the image of the laser line appears on the image plane of the CCD camera. According to the depth of object the position of captured laser line in the image is changed.

3.2 SLAM with the ICP algorithm

By using the depth information and the ICP algorithm, we can apply the SLAM algorithm. (Se et al. 2002) Here we will briefly introduce this ICP algorithm. (Pulli 1997; Rusinkiewicz et al. 2001; Sharp et al. 2002; Zhang 2002; Chetverikov et al. 2005; Li et al. 2003; Shibata et al. 2003) First, in the original ICP algorithm, we regard the sets of points that compose the range image of the previous pose as M and the sets of points in the range image of the current pose as L (as shown in Fig. 2). The alignment process works to minimize the mean squared distance between the scene points and their closest model point. For each point, L_i, from the set L, there is at least one point that is closer to L_i than all the other points in M. This point, M_i, is defined as the closest point. After determining the closest points, we then need to find a 3-D transformation which, when applied to the data set L, minimizes the distance between two point sets. In short, the goal of the problem can be expressed as follows:

$$\min \sum_{1}^{N} \|M_i - (RL_i + T)\|^2 \tag{1}$$

where R is a 3-by-3 rotation matrix, T is a 3-by-1 translation vector, and the subscript i refers to the corresponding elements of sets M and L. To solve R and T, we can either use the unit quanternions or the singular value decomposition method.

The procedures of the ICP algorithm can be stated as follows:

Step 1: For each point in L, compute the closest point in M.

Step 2: With the closest point pairs computed in step one, compute the translation matrix, T, and the rotation matrix, R, by using the least square method (equation (4)), where the total number of points in L is N.

$$\left\{ \begin{matrix} S = \sum_{1}^{N} \|M_i - (RL_i + T)\|^2 \\ \min(S) \end{matrix} \right\} \tag{2}$$

Step 3: Apply the transformation matrix T and R to all the points in L.

$$L_i^{'} = RL_i + T \tag{3}$$

Step 4: Compute the mean square error of each closest points pairs.

$$d_i = \|M_i - (RL_i + T)\|^2 \tag{4}$$

Step 5: Compute the change in the total mean square error, which is the sum of the mean square error of each point pairs.

$$S = \sum_i d_i = \sum_i \|M_i - (RL_i + T)\|^2 \tag{5}$$

Step 6: If the change in error is less than a threshold value (< threshold), then the algorithm ends. If not, go to step 1.

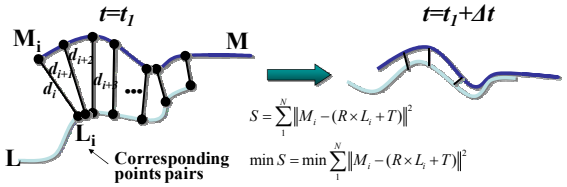


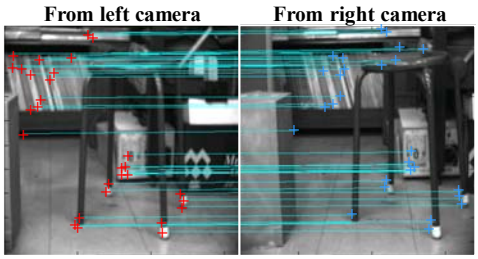
Fig. 2. Concept of the ICP algorithm

4. SCALE-INVARIANT FEATURE TRANSFORM

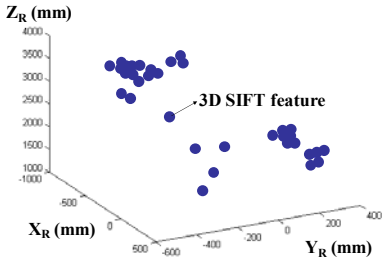
4.1 SIFT feature point extraction

The SIFT feature is a kind of robust feature point that can be extracted from a 2-D image. (Lowe 1999; Lowe 2004) Each SIFT feature has a descriptor, which is composed of 128 parameters, and in most cases, the descriptor is unique to each SIFT point. The SIFT features are therefore invariant to image scaling and rotation, and partially invariant to changes in illumination and to a 3-D camera viewpoint. Being well localized in the spatial and frequency domains, they also reduce the probability of disruption by occlusion, clutter, or noise. Large numbers of features can be extracted from

typical images with efficient algorithms. In addition, because the features are highly distinctive, they enable a single feature to be correctly matched with a high probability against a large database of features and they provide a basis for scene recognition and mobile robot localization.



(a) SIFT feature matching



(b) Feature map

Fig. 3. Feature matching with the SIFT

4.2 SIFT Stereo

We compute 3-D world coordinates for each feature by matching pairs of images from two stereo cameras. Using the epipolar and SIFT feature invariance constraint, we can match features in the right-left image pairs. As shown in Fig. 3(b), the SIFT features and their 3-D coordinates serve as landmarks for the self-localization as well as for the local map alignment and matching of the mobile robot.

4.3 Local Map Alignment with the SIFT

To build a map, we need to know how the robot moves between frames. Robot odometry can give a rough estimate, though it is prone to errors such as slipping. Nonetheless, it enables us to more efficiently predict the region in which we need to search for each match. Once the SIFT features are matched, we can obtain the 3-D coordinate of the SIFT features. To this end, we can use the matches between a previous view (namely the view of the robot at a previous position) and the current view (namely the view of the robot at the current position) in a least square procedure to compute a more accurate camera ego-motion and, hence, better localization.

Many researchers have tested and proved the robustness of localization using SIFT features. As indicated, however, the SIFT algorithm cannot work if only a few SIFT intensity features exist in the environment. Furthermore, a map built by the SIFT cannot be used for obstacle avoidance and next view generation. We therefore propose to fuse the SIFT algorithm with the TrICP algorithm, and to consequently integrate the SIFT feature map built by the SIFT with the laser map built by the TrICP algorithm.

5. SENSOR FUSION ALGORITHM

In this chapter, we introduce our fusion algorithm. (Lee et al. 2005) The basic idea of the fusion algorithm is to fuse the information obtained by both the SIFT and the TrICP algorithm.

5.1 Dempster Shafer Sensor Fusion Algorithm

If we estimate the probability of two events A and B using two sensors, then, given two sensors with probability masses of P1(A) (the probability of A being estimated by sensor 1), P1(B) (the probability of B being estimated by sensor 1), P2(A) (the probability of A being estimated by sensor 2), P2(B) (the probability of B being estimated by sensor 2), U1 (the uncertainty of sensor 1), and U2 (the uncertainty of sensor 2), we can describe the Dempster Shafer sensor fusion algorithm as follows:

$$P_{fusion}(A) = \frac{P_1(A)P_2(A) + P_1(A)U_2 + U_1P_2(A)}{k} \quad (6)$$

$$P_{fusion}(B) = \frac{P_1(B)P_2(B) + P_1(B)U_2 + U_1P_2(B)}{k}$$

$$k = 1 - P_1(A)P_2(B) - P_1(B)P_2(A)$$

where P_{fusion} refers to the probability estimate of each event after fusion and k is the normalization parameter.

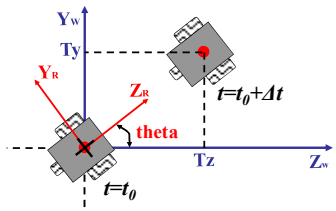


Fig. 4. Axis of the robot's movement

5.2 Sensor Fusion Algorithm for Self-Localization

As mentioned, our method of mobile robot self-localization and mapping relies on both the TrICP algorithm and the SIFT, respectively. Fig. 4 shows the axis of the robot. At each step, our mobile robot has only two kinds of movement. The first movement is the translation along the Z coordinate, and the second is the rotation around its local origin. Hence, the main task of our fusion algorithm is to obtain the translation in the Z direction, T_z, as well as the rotation theta.

Step 1: First we use the trimmed ICP method to obtain the robot translation in the Z direction, Tz, as well as the translation in the y direction, Ty, and the robot rotation angle, theta. Next, we denote the Tz estimated by the TrICP algorithm as Tz_icp; we also denote the Ty estimated by the TrICP algorithm as Ty_icp, and we denote the theta estimated by the TrICP algorithm as theta_icp. By using the SIFT to implement the SIFT feature extraction and self-

localization, we can estimate Tz, Ty, and theta on the basis of the estimation of the SIFT feature points matching. Similarly, we denote the Tz estimated by the SIFT as Tz_sift; we denote the Ty estimated by the SIFT as Ty_sift; and we denote the theta estimated by the SIFT as theta_sift. In fact, neither Tz_icp, Ty_icp and theta_icp nor Tz_sift, Ty_sift and theta_sift are likely to be the exact value of Tz, Ty, and theta. To obtain a more accurate value, we therefore need a sensor fusion algorithm.

Step 2: Once we get an estimate of the robot's position from the TrICP algorithm (Position_icp) and the SIFT (Position_sift), we can assume that the real position of the robot might be near to the two positions estimated by the TrICP algorithm and the SIFT. Subsequently, we can generate several candidate positions in the region near Postion_icp and Position_sift. To do this, we assume that Tz_candidate(i) is the candidate value that we have generated for the real value of Tz, and that one of the values of Tz_candidate(i) must be closest to the real value of Tz. Similarly, we assume that theta_candidate(k) is the candidate value we generated for the real value of theta. Here, i = 1~N and k = 1~N (where N is the number of candidate values). In addition, we did not calculate the fusion results for Ty because of the likelihood of Ty being small in most of the navigation cases (since the robot only moves in the Z direction). We therefore calculated Ty by combining Ty_icp and Ty_sift.

Step 3: Once we obtain the candidate position values, we assign a probability value to each value of T_z_candidate(i) and theta_candidate(k). First, we propose to assign a probability value on the basis of the TrICP algorithm. To assign the probability value, we should first obtain the probability distribution curve of the estimation error of Tz_icp and theta_icp. Obtaining an exact probability distribution curve is not easy; hence, researchers usually obtain it through experimentation or from experience. Of course, the greater the number of experiments, the greater the accuracy of the probability curve. In our case, we conducted 60 experiments, each time using different objects in the environment. Furthermore, we limited the mobile robot's movement to only 300 mm to 800 mm in the Z direction for each step, along with a rotation range of 10 degrees to 40 degrees for each step. The reason for the limitation is that these ranges are generally adequate for robot navigation.

In our experiment, the probability distribution curve of the estimation error of Tz_icp is similar to the Gaussian curve. We therefore approximated the probability distribution curve as a Gaussian distribution with a mean of 16 mm and a deviation of 6 mm. Similarly, we approximated the probability curve for theta_icp as a Gaussian curve with a mean of 0.2 degrees and a deviation of 2 degrees. After obtaining the probability distribution, we then assigned the probability for each candidate position.

Step 4: Similarly, through experimentation, we can also get the following: a probability distribution curve of the estimation error of Tz_sift, a Gaussian distribution with a mean of 7 mm and a deviation of 10 mm, a probability distribution curve of the estimation error of theta_sift, a Gaussian distribution with a mean of 0.1 degrees and a deviation of 1.7 degrees. Furthermore, the uncertainty value of the SIFT estimation is based on the number of matched SIFT feature points. Our experimentation also shows that the greater the number of SIFT feature points, the greater the accuracy of the estimation results. Actually, if only a few SIFT features exist in the environment, the uncertainty of the SIFT estimation is very high. Table 2 shows the uncertainty assignment for the SIFT estimation.

Table 1. Uncertainty assignment for the TrICP algorithm

Number of corners	Uncertainty (U)
0	U = 0.8
1	U = 0.5
≥2	U = (L-x)/L×0.5 (x ≤ 1/2L) U = 0.2 (x > 1/2L) L = 200 mm (x is the length of the overlapping region)

Step 5: Up to this stage, we have obtained the probability and uncertainty assignment from the TrICP algorithm and from the SIFT. We can therefore obtain the probability value of each candidate position on the basis of the fusion. Consequently, the candidate position with the highest probability is the position we would choose for the mobile robot.

Table 2. Uncertainty assignment for the SIFT

Number of SIFT features	Uncertainty (U)
≤ 3	U = 0.9
≤ 7	U = 0.8
≤ 12	U = 0.6
≤ 17	U = 0.5
≤ 22	U = 0.4
> 22	U = 0.2

6. EXPERIMENTAL RESULTS

6.1 Parameter Analysis of Sensor Fusion

Through our experiments, we found that the number of overlapping points between the previous laser range data and the current laser range data, as detected by the robot, can influence the self-localization results of the TrICP algorithm; we also found that the number of SIFT intensity features in the environment can influence the self-localization results of the SIFT. Consequently, the two parameters influence the fusion results. We therefore analyze the two parameters in this section.

First we made the mobile robot move in the Z_R direction. For each movement of the robot, we obtained the real translation distance in the Z_R direction, Transz_real; we also used the TrICP algorithm to estimate the robot's movement in the Z_R direction, Transz_icp; similar estimations were made with the SIFT for Transz_sift and with fusion for Transz_fusion. Consequently, we define the error of self-localization for one step as follows:

$$\text{Erroricp} = \text{Transz\_icp} - \text{Transz\_real} \quad (7)$$
$$\text{Errorsift} = \text{Transz\_sift} - \text{Transz\_real}$$
$$\text{Errorfusion} = \text{Transz\_fusion} - \text{Transz\_real}.$$

By keeping the number of SIFT features constant in the environment, we changed the number of overlapping points between the previous laser range data and the current laser range data, all of which could be detected by the mobile robot with the aid of a laser structured light range finder. For each environment, we conducted the experiments 5 times, and we obtained the mean value and the deviation of the experimental data.

Fig. 5 shows the accuracy and uncertainty of the TrICP-based self-localization results with the change of overlapping points between the previous laser range data and the current laser range data detected by the robot in the environment. We can see that whenever the number of overlapping points is small the localization accuracy is low and the estimation uncertainty is quite high. Moreover, with the increase in the number of overlapping points between the two range data sets detected from the environment, the error and uncertainty of the estimation tend to be lower.

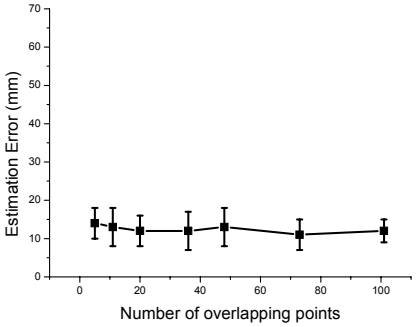


Fig. 5. Results of the TrICP-based self-localization for analysis of the number of overlapping points

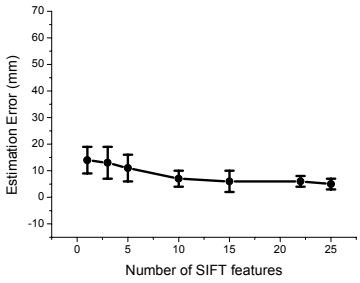


Fig. 6. Results of the SIFT based self-localization for the analysis of the number of SIFT features

Fig. 6 shows the accuracy and uncertainty of the SIFT-based self-localization results with the change in the number of SIFT features the robot can detect in the environment. We can see that whenever the number of SIFT features detected by the robot is less than 5 the localization accuracy is low and the estimation uncertainty is quite high. Moreover, with the increase in the number of SIFT features detected from the environment, the error and uncertainty of the estimation tend to be lower.

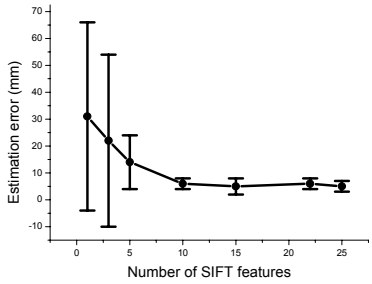


Fig. 7. Results of the fusion experiment for the analysis of the number of SIFT features

Fig. 7 shows the accuracy and uncertainty of the fusion-based self-localization results with the change of SIFT features the robot can detect in the environment. We can see that the error and uncertainty of the estimation tend to be lower. However, in contrast to the estimation results of the SIFT, the fusion estimation results tend to produce a high level of accuracy and a low degree of uncertainty, regardless of the number of SIFT features. This phenomenon implies that sensor fusion, in comparison with the SIFT method, tends to produce self-localization results with a lower degree of uncertainty.

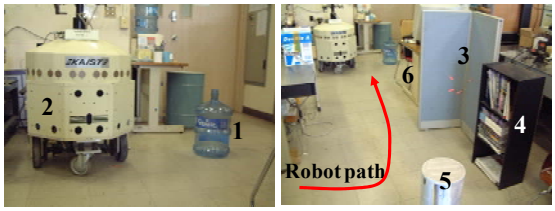


Fig. 8. Environment for the mobile robot's navigation

6.2 Sensor Fusion Experiment in a Real Environment

Fig. 8 shows the objects in the experimental environment. The objects include a chair, a shelf, books, a panel, a table, an experimental device, and various polygonal objects, all of which are commonly seen in our lab environment. Fig. 9 shows the map that was built includes laser structured line data and SIFT feature points. The results of the map alignment are acceptable as shown in Fig. 9 (a) and (b). However, if we only use the TrICP algorithm and the SIFT algorithm in the robot self-localization and map alignment process, as shown in Fig. 9 (c) and (d), the results of the map alignment are inaccurate in certain areas. This inaccuracy shows that the localization and map matching results have a high degree of uncertainty if there is no fusion and only a single method is used.

6. CONCLUSIONS

Self-localization and the matching between two range images are important and challenging requirements for autonomous mobile robot navigation. We propose a sensor fusion algorithm based on the Dempster Shafer method. This

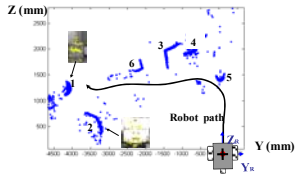
algorithm makes use of the advantage of the TrICP algorithm and the SIFT to produce better localization and mapping results.

1). If the overlapping points between the previous laser range data and the current laser range data are increased with a fixed number of SIFT features, the uncertainty of robot localization is decreased and the accuracy of the fusion algorithm is increased.

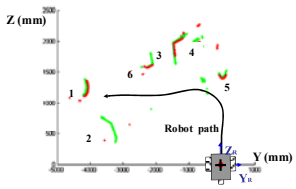
2). If the SIFT feature points between the previous image data and the current image data are increased with a fixed number of overlapped laser range data points, the uncertainty of robot localization is decreased and the accuracy of the fusion algorithm is increased.

Our experimental results in real robot navigation environments confirm the usefulness and robustness of the proposed method.

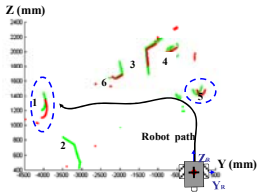
In our future work, we plan to refine the proposed algorithm by doing more experiments in various kinds of environments in order to strengthen its robustness in more complex environments and to improve the accuracy of self-localization on uneven ground.



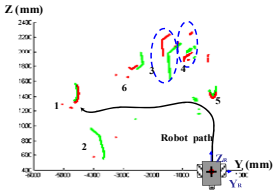
(a) map alignment using Fusion – SIFT feature map



(b) map alignment using Fusion – laser map



(c) map alignment using TrICP – laser map



(d) map alignment using SIFT – laser map

Fig. 9. The map comparison

REFERENCES

Cho, H.S. (2005) *Optomechatronic: Fusion of Optical and Mechatronic Engineering*, CRC Press.

Dixon, J. and O. Henlich (1997) Mobile Robot Navigation http://www.doc.ic.ac.uk/~nd/surprise_97/journal/vol4/jmd/.

Surmann, H., A. Nucheter and J. Hertzberg, (2003) An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments, *Robotics and Autonomous Systems* **45**:181-198.

Downs, A., R. Madhavan and T. Hong, (2003) Registration of Range Data from Unmanned Aerial and Ground Vehicles, *In Proc. the 32nd Applied Imagery Pattern Recognition Workshop*, pp. 45-50.

Besl, P. and N. McKay, (1992) A method for registration of 3-D shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**: 239-256.

Pulli,K. (1997) Surface reconstruction and display from range and color data, *PhD thesis, University of Washington, Seattle*.

Rusinkiewicz, S. and M. Levoy, (2001) Efficient variants of the icp algorithm, *In Proc. Third International Conference on 3-D Digital Imaging and Modeling*, pp.145-152.

Sharp, G.C., S.W. Lee and D.K. Wehe, (2002) ICP Registration Using Invariant Features, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**: 90-102.

Zhang, Z. (1994) Iterative point matching for registration of free-form curves and surfaces, *International Journal of Computer Vision*, **13**: 119-152.

Chetverikov, D., D. Stepanov and P. Krsek, (2005) Robust Euclidean alignment of 3D point sets: the trimmed iterative closest point algorithm, *Image and Vision Computing* **23**: 299-309.

Li, B. and H. Holstein, (2003) Using k-d Trees for Robust 3d point pattern Matching, *In Proc. the Fourth International Conference on 3-D Digital Imaging and Modeling*, pp. 95-102.

Shibata, T., T. Kato and T. Wada (2003) K-D Decision Tree: An Accelerated and Memory Efficient Nearest Neighbor Classifier, *In Proc. the Third IEEE International Conference on Data Mining*, pp. 641-644.

Lee, H.K., S. Xingyong, M.Y. Kim, M.Y and H.S. Cho (2005) A Novel Robot Sensor System Utilizing the Combination Of Stereo Image Intensity And Laser Structured Light Image Information, *In Proc. International Conference on control, Automation and Systems*, KINTEX, Korea, pp. 729-734.

Izquierdo, M.A.G., and M.T. Sanchez, M.T. (1999) Sub-pixel measurement of 3D surfaces by laser scanning, *Sensors and Actuators*, **76**: 1-8.

Se, S., D. Lowe and J. Little (2002) Vision-based Mapping with Backward Correction, *In Proc. the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, pp. 153-158.

Lowe, D.G. (2004) Distinctive Image Features from Scale Invariant Keypoints, *International Journal of Computer Vision* **60**: 91-110.

Lowe, D. (1999) Object Recognition from Local Scale-Invariant Features, *In Proc. Int. Conference on Computer Vision*, pp. 1-8.