Sensors and Materials, Vol. 26, No. 5 (2014) 353–364
MYU Tokyo

S & M 1002

Robot Simultaneous Localization and Mapping Using a Calibrated Kinect Sensor

Chen-Tung Chi[1,2], Yin-Tien Wang[2,*], Shao-Ting Cheng[2] and Chin-An Shen[2]

[1]Department of Mechanical Engineering, Taipei Chengshih University of Science and Technology, Taipei 112, Taiwan
[2]Department of Mechanical and Electro-Mechanical Engineering, Tamkang University, New Taipei City 251, Taiwan

(Received December 24, 2013; accepted March 6, 2014)

Key words: RGB-D sensor, sensor calibration, simultaneous localization and mapping (SLAM), visual mapping

In this paper, we present an algorithm for robot simultaneous localization and mapping (SLAM) using a Kinect sensor, which is a red-green-blue and depth (RGB-D) sensor. The distortions of the RGB and depth images are calibrated before the sensor is used as a measuring device for robot navigation. The calibration procedure includes the correction of the RGB image as well as alignment of the RGB lens with the depth lens. In SLAM tasks, the speeded-up robust features (SURFs) are detected from the RGB image and used as landmarks for building the environmental map. The depth image further provides the stereo information to initialize the three-dimensional coordinates of each landmark. Meanwhile, the robot estimates its own state and landmark locations using the extended Kalman filter (EKF). Two SLAM experiments were carried out in this study and the results showed that the Kinect sensors could provide reliable measurement information for mobile robots navigating in unknown environments.

1. Introduction

When a robot is navigating in an unknown environment, it relies on sensors to recognize the outside world and then to estimate the state of the robot itself to achieve the task of autonomous navigation. Commonly used sensors include the laser range finder (LRF) and vision sensor. The LRF can offer high-precision measurement data, but it is too expensive to be extensively used. The vision sensor has a relatively wide range of cost, from low-cost low-end to expensive high-order products, being generally applied in a robot's sensing devices. However, vision sensors capture only two-dimensional images that have no depth information of environmental objects. One must perform expensive calculation to recover the depth information for visual measurement.

*Corresponding author: e-mail: ytwang@mail.tku.edu.tw

Recently, Microsoft has released a red-green-blue and depth (RGB-D) sensor called Kinect, which has an RGB image sensor and a depth sensor.[1] The RGB sensor captures color images of the environment. The depth sensor uses an infrared transmitter and a complementary metal oxide semiconductor (CMOS) camcorder to detect the depth of the corresponding objects in RGB images. When infrared light reaches rough objects or penetrates through frosted glass, the reflection spots or random scatterings, called laser speckle, are measured using the CMOS camcorder. The depth information is then generated by recording the laser speckle at different positions and distances, followed by comparative and statistical analyses.[2] The Microsoft Kinect sensor has several advantages, including low price, multiple sensing capability, and the availability of a free software development toolkit. Such a sensor is suitable for 3D modeling[3] and robot navigation in the environment because of its functions and low price. This study uses Kinect to capture color and depth images as environmental information for mobile robot navigation.

The distortions of RGB and depth images need to be calibrated before Kinect is applied as a measuring device. The distortion of Kinect's RGB image depends on the lens structure of the camera. Many procedures have been developed to calibrate the RGB image by determining the intrinsic parameters of the camera.[4–6] In these procedures, the camera or objects were arbitrarily moved and the conversion relationship of different images was obtained to determine the camera's intrinsic parameters and external transformation parameters. Zhang carried out a complete derivation of the conversion relationship between the images and objects, and then used the Levenberg-Marquardt optimization method to obtain accurate parameters.[4] Heikkila and Silven established the image correction model to obtain the corresponding parameters.[5] A well-known Matlab toolbox based on these calibration algorithms has been released.[6] In this study, we use these algorithms and toolbox to deal with the calibration of the RGB images. On the other hand, the distortion of Kinect's depth image is due to the misalignment of the RGB sensor against the depth sensor. However, there is no popular method of aligning the lenses of these two sensors.[7] In this study, we propose a novel procedure to transfer and align the depth image with the RGB image. The transformation of the sensor location is modeled using a mathematical interpolation model. We choose the origin of the RGB camera as the reference frame and then transform the depth image to align it with the RGB camera frame.

The image features detected from Kinect RGB images can be used to represent the landmarks in the environment and build an environmental map for robot navigation. A detection method based on the scale-invariant feature was developed by Lindeberg.[8] He established image Hessian matrices, whose elements are the convolution of the Gaussian second-order derivative with the image. An image feature is selected by examining the determinant of the Hessian matrix based on the nonmaximum suppression rule. The scale-invariant features have the advantages of requiring high stability and repeatability. On the other hand, these features have the disadvantage of extensive computation. Concerning the issue of computational speed, Bay *et al*. replaced the Gaussian second-order derivative with the box filter, and calculated the approximation of determinant of the Hessian matrix using the integral image method.[9] This method, called speeded-up robust features (SURFs), significantly reduces the calculation time. In this study, the SURF algorithm is employed to detect the features from Kinect RGB images and to

represent the landmarks in the environmental map. To initialize the three-dimensional coordinates of each landmark, the information of Kinect's depth image is integrated with the pixel coordinates of the corresponding SURF feature. Meanwhile, in this study, the extended Kalman filter (EKF)[10] is used to recursively predict and estimate the robot state as well as the environmental landmarks.

In this study, we propose an algorithm for robot SLAM using calibrated Kinect RGB-D sensors. The algorithm was validated in the experiments by performing the SLAM tasks on an actual system using a Kinect system as the only sensing device. The contributions of this study are novel algorithms for solving the problem of aligning depth images with color images as well as for calibrating the Kinect RGB sensor. In this study, we also extend the usability of local invariant feature detectors in SLAM tasks by utilizing their robust representation of visual landmarks. Data association and map management for SURF-based mapping are also developed to improve the robustness of SLAM systems.

The remaining sections of the paper are organized as follows. Section 2 presents the calibration methods of RGB-D sensors. The structure of the developed robot SLAM system and the method of SURF-based mapping are described in §§ 3 and 4, respectively. Section 5 depicts two experimental applications and results of the proposed algorithms. Finally, concluding remarks are given in the last section.

2. Calibration of RGB-D Sensors

In this study, we develop novel algorithms to align depth and color images as well as to calibrate the Kinect RGB sensor. The concept and procedures of the developed algorithms are described in the following subsections.

2.1 *Alignment of RGB and depth images*

Kinect captures both color and depth images, as shown in Figs. 1(a) and 1(b), in order to construct the stereo coordinates of environmental objects. However, the pixel coordinates of a corresponding object in these two images could not be matched correctly because the RGB and depth cameras are not located at the same position. There exists an unknown transformation between the RGB and depth images. We need to determine the relationship between these two image planes to correct the noticeable image slant. However, some defects exist in the depth image and the depth information

(a) (b) (c)

Fig. 1. (Color online) (a) RGB image, (b) depth image, and (c) infrared image.

is not available in these defective areas, as shown in Fig. 1(b). Therefore, a one-to-one relationship is not guaranteed between the RGB and depth images. On the other hand, the original infrared image does not contain any defective areas, as shown in Fig. 1(c). The depth image is a grayscale image converted from the depth datum of the original infrared image. Therefore, we can align the RGB image with the infrared image, which has the same coordinates as those of the depth image. The alignment procedure is implemented using geometrical transformation.[11] In this study, we use a pair of bilinear equations to represent the pixel coordinate transformation between RGB and infrared images,

$$x' = Ax + By + Cxy + D, \tag{1}$$

$$y' = Ex + Fy + Gxy + H, \tag{2}$$

where x and y are the coordinates on the RGB image and x' and y' are the corresponding coordinates on the infrared image. The eight coefficients (A to H) can be determined by the least-squares method if there are more than four known corresponding points. The transformation model with determined coefficients can be used to transform all the pixels within the quadrangle area defined by the four vertices. By substituting the pixel coordinates of the features in the RGB image into eqs. (1) and (2), the pixel coordinates of features in the infrared image can be determined, and the corresponding depth information of the feature can be obtained. The alignment and transformation results are depicted as circle marks in Fig. 2. The figure shows the root-mean-square error (RMSE) of image alignments with respect to different numbers of corresponding feature points. The feature number 0 indicates the original situation without alignment, which has a RMSE of 16.05 pixels. When using four corresponding points for alignment, the RMSE reduces to 2.18 pixels. The RMSE does not change considerably when the feature number varies from 4 to 60. In this study, we also use a pair of nonlinear equations to represent the pixel coordinate transformation as

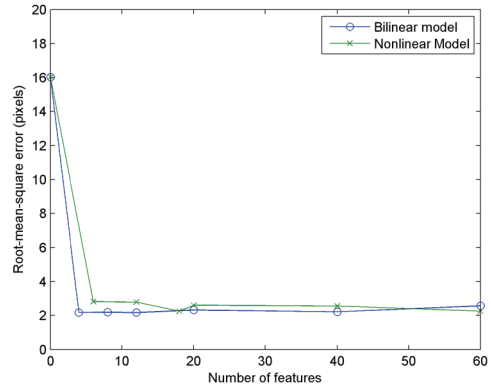


Fig. 2. (Color online) RMSE of alignment results using varying numbers of corresponding features.

$$x' = Ax + By + Cxy + Dx^2 + Ey^2 + F, \tag{3}$$

$$y' = Gx + Hy + Ixy + Jx^2 + Ky^2 + L. \tag{4}$$

The alignment and transformation results are depicted as cross marks in Fig. 2. The nonlinear model does not provide a better RMSE than the bilinear model. Therefore, in this study, we chose the bilinear model with four corresponding points as the transformation model.

2.2 *Calibration of RGB image*

In this study, we use the algorithms and Matlab toolbox in refs. 4−6 to calibrate the Kinect RGB images. A total of 20 photos of a chess board are taken, as shown in Fig. 3. Each photo is captured in a different orientation with horizontal rotation angle always less than 90°. The study uses the image distortion model[4] to represent the lens distortion as

$$\begin{bmatrix} \frac{h_{dx}^C}{h_{dz}^C} \\ \frac{h_{dy}^C}{h_{dz}^C} \end{bmatrix} = (1 + k_d(1)r^2 + k_d(2)r^4) \begin{bmatrix} \frac{h_x^C}{h_z^C} \\ \frac{h_y^C}{h_z^C} \end{bmatrix} + \begin{bmatrix} 2k_d(3)\frac{h_x^C}{h_z^C}\frac{h_y^C}{h_z^C} + k_d(4)\left(r^2 + 2\left(\frac{h_x^C}{h_z^C}\right)^2\right) \\ k_d(3)\left(r^2 + 2\left(\frac{h_y^C}{h_z^C}\right)^2\right) + 2k_d(4)\frac{h_x^C}{h_z^C}\frac{h_y^C}{h_z^C} \end{bmatrix}, \tag{5}$$

$$r = \sqrt{\left(\frac{h_x^C}{h_z^C}\right)^2 + \left(\frac{h_y^C}{h_z^C}\right)^2}, \tag{6}$$

where $\boldsymbol{k}_d$ is the vector of distortion coefficients of the RGB camera, $\begin{bmatrix} \frac{h_x^C}{h_z^C} & \frac{h_y^C}{h_z^C} \end{bmatrix}^T$ is the normalized image coordinate in three-dimensional space, $\boldsymbol{h} = [h_x^C \; h_y^C \; h_z^C]^T$ is the undistorted ray vector, and $\boldsymbol{h}_d$ is the corresponding distorted ray vector. In practice, the

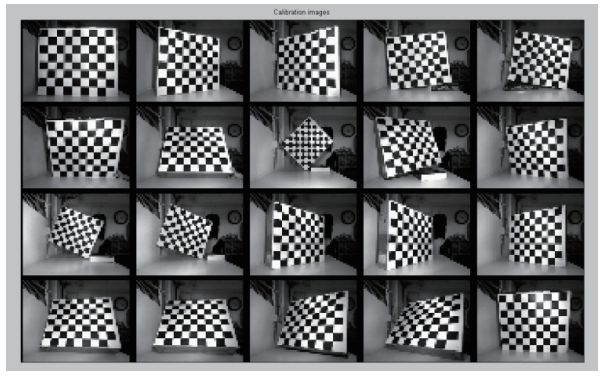


Fig. 3. Twenty photos of a chess board.

distorted ray vectors are available and the undistorted ray vectors are to be determined. In the study, we use the image correction model[5] to represent the inversion of the image distortion model as

$$\begin{bmatrix} \frac{h_x^C}{h_z^C} \\ \frac{h_y^C}{h_z^C} \end{bmatrix} = \begin{bmatrix} \frac{h_{dx}^C}{h_{dz}^C} \\ \frac{h_{dy}^C}{h_{dz}^C} \end{bmatrix} + \frac{1}{G} \begin{bmatrix} \left(k_c(1)r_d^2 + k_c(2)r_d^4\right)\frac{h_{dx}^C}{h_{dz}^C} + 2k_c(3)\frac{h_{dx}^C}{h_{dz}^C}\frac{h_{dy}^C}{h_{dz}^C} + k_c(4)\left(r_d^2 + 2\left(\frac{h_{dx}^C}{h_{dz}^C}\right)^2\right) \\ \left(k_c(1)r_d^2 + k_c(2)r_d^4\right)\frac{h_{dy}^C}{h_{dz}^C} + k_c(3)\left(r_d^2 + 2\left(\frac{h_{dy}^C}{h_{dz}^C}\right)^2\right) + 2k_c(4)\frac{h_{dx}^C}{h_{dz}^C}\frac{h_{dy}^C}{h_{dz}^C} \end{bmatrix}, \quad (7)$$

$$G = 4k_c(5)r_d^2 + 6k_c(6)r_d^4 + 8k_c(7)\frac{h_{dy}^C}{h_{dz}^C} + 8k_c(8)\frac{h_{dx}^C}{h_{dz}^C} + 1, \quad (8)$$

where $\boldsymbol{k}_c$ is the vector of correction coefficients. Perspective projection[12] is used to model the transformation of the 3D space coordinate system to a 2D image plane. An equivalent image plane is established to represent the ray vector and the measurement vector of the ith observed image feature as

$$\frac{h_{dx}^C}{h_{dz}^C} = \frac{(I_{dx} - u_0)}{f_u}; \frac{h_{dy}^C}{h_{dz}^C} = \frac{(I_{dy} - v_0)}{f_v}. \quad (9)$$

Focal lengths f_u and f_v denote the distance from the camera lens center to the image plane along the u- and v-axes, respectively, (u_0, v_0) is the offset pixel vector of the image plane, and (I_{dx}, I_{dy}) are the pixel coordinates of a measured feature in the image plane. The Matlab toolbox[6] determines the intrinsic parameters and distortion coefficients, which are listed in Table 1. The correction parameters $\boldsymbol{k}_c$ in Table 2 are determined by reversing the image distortion model.

Table 1
Image distortion coefficients.

Focal length	f_u	260.09598 ± 4.02164 (pixel)
	f_v	261.04304 ± 4.04892 (pixel)
Principal point	u_0	166.32693 ± 4.28107 (pixel)
	v_0	126.76935 ± 3.35258 (pixel)
Distortion coefficient	$k_d(1)$	0.24376 ± 0.04412
	$k_d(2)$	−0.70139 ± 0.20824
	$k_d(3)$	−0.00194 ± 0.00693
	$k_d(4)$	0.01339 ± 0.00839

Table 2
Image correction coefficients.

$k_c(1)$	$k_c(2)$	$k_c(3)$	$k_c(4)$	$k_c(5)$	$k_c(6)$	$k_c(7)$	$k_c(8)$
−0.2479	0.7104	0.0018	−0.0133	0.2800	−0.8036	−0.0005	0.0158

3. Robot SLAM

When the robot performs SLAM tasks, the states of the robot and landmarks in the environment are estimated on the basis of measurement information. The state sequence of a system at time step k can be expressed as

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}, w_{k-1}), \tag{10}$$

where $\boldsymbol{x}_k$ is the state vector, $\boldsymbol{u}_k$ is the input, and w_k is the process noise. When performing SLAM tasks using a Kinect sensor, the state vector contains the states of the sensor and landmarks,

$$\boldsymbol{x} = [\boldsymbol{x}_C^T, \boldsymbol{M}^T]^T = [\boldsymbol{x}_C^T, \boldsymbol{m}_1^T, \boldsymbol{m}_2^T, \cdots, \boldsymbol{m}_j^T]^T, \tag{11}$$

where $\boldsymbol{x}_C = [r^T, \varphi^T, v^T, \omega^T]^T$ denotes the sensor coordinates in world frame, and $\boldsymbol{m}_j$ represents the jth landmark in the environmental map $\boldsymbol{M}$. The objective of the robot SLAM tasks is to estimate the state $\boldsymbol{x}_k$ of the target recursively according to the measurement $\boldsymbol{z}_k$ at k,

$$\boldsymbol{z}_k = g(\boldsymbol{x}_k, v_k), \tag{12}$$

where v_k is the measurement noise. A Kinect sensor system is the only sensing device considered in the recursive state estimation algorithm. At time $t = k$, the vectors of measurement $\boldsymbol{z}_k$ and the ith observed image feature are, respectively,

$$\boldsymbol{z}_k = [\boldsymbol{z}_{1k}^T, \boldsymbol{z}_{2k}^T, \cdots, \boldsymbol{z}_{mk}^T]^T, \tag{13}$$

$$\boldsymbol{z}_{ik} = \begin{bmatrix} I_{ix} \\ I_{iy} \end{bmatrix}, \tag{14}$$

where $i = 1, 2, \ldots, m$; m is the number of measurements at time k. Since the sensor frame is set at the center of the RGB camera lens, the coordinate representation in the depth image is transformed to the RGB camera. The ith observed image feature can then be initialized using the 3D coordinates in the world frame (Fig. 4) as

$$\boldsymbol{m}_i = \boldsymbol{r} + \boldsymbol{R}\boldsymbol{h}_i^C, \tag{15}$$

where $\boldsymbol{r}$ is the position vector of the sensor frame, $\boldsymbol{R}$ is the rotational matrix[13] from the world frame to the sensor frame, and $\boldsymbol{h}_i^C$ is the ray vector of the image features in the sensor frame obtained from eq. (9). In this study, the handheld Kinect sensor is the only sensing device used for measurement in the SLAM system. The handheld Kinect sensor is treated as a free-moving robot system with unknown inputs.[14] System states are estimated using the EKF estimator to solve the target tracking problem.[14,15]

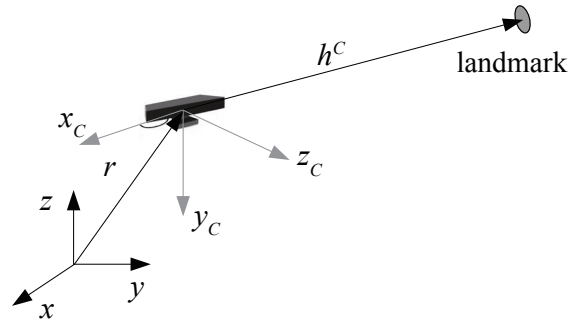


Fig. 4. Kinect sensor system.

4. SURF-Based Mapping

Mapping in visual SLAM requires a robust method of representing visual landmarks detected in an image. In this study, we used the SURF method to detect and represent visual landmarks of a map during SLAM tasks. The SURF method developed by Bay *et al*. uses a box filter instead of a difference of Gaussians to approximate the determinant of the Hessian matrix.[9] Feature detection is performed at the pixel location where the determinant of the Hessian matrix is a local maximum value. The box filter is further combined with the integral image method[16] to reduce the image processing time. Bay *et al*. applied the octaves concept to the design of box filter sizes for detecting features in different scales. After the features are detected from the image, the description vector is computed to represent feature characteristics. Bay *et al*. described the orientation of a feature using a Haar wavelet filter to compute the wavelet responses of the image feature.[9] The orientation of a feature is defined as the direction with the largest sum of the Haar wavelet responses. A high-dimensional description vector is then used to describe the uniqueness of the feature. For matching high-dimensional description vectors, the most popular method is the nearest-neighbor (NN) search method.[17] For a space of n dimensions with point sets P and Q, the query point q belongs to Q. The NN search method can be used to find the point p in P, which has a minimum distance from q. The distance between the arbitrary point p in the P set and the query point q is usually defined by the norm l_s. In the case of l_2, the distance becomes the Euclidean distance d. The criterion for matching two image features is usually to determine the shortest distance between their descriptors.

The implementation of the SLAM system with a Kinect sensor integrates the motion model, the measurement model, and the SURF detection algorithm. Once the images are captured by the Kinect RGB camera, feature extraction is performed by the SURF method. The system performs data association of the landmarks in the map database and the image features of the extracted SURF using the proposed matching criterion. A map management system is also designed to coordinate the newly extracted features and the "bad" features in the system. After the properties of newly extracted features are investigated, a detection algorithm is used to distinguish moving and stationary objects.

The state variables of all stationary landmarks are augmented in the state vector in eq. (10). However, features that are not continuously detected at each time step are considered as "bad" features and are deleted from the state vector.

5. Experimental Results

Two experiments including ground truth and robot SLAM are carried out to validate the proposed algorithms.

5.1 *Ground truth*

The first experiment is a ground truth experiment to test the performance of SLAM using the Kinect sensor. The sensor is carried to follow a 1 m^2 track, as shown in Fig. 5. The sensor lens is directed toward the opposite wall in order to capture the image features. There were three detour laps around the square track and about 2000 images were captured in each circulation. The experimental results of ground truth are listed in Table 3. The first three columns of the table are ground truth locations of ground truth points, and the last three columns are the estimated locations of those points. The ± variation values indicate the standard deviations in the estimated values. The experimental results show that the developed EKF-SLAM system with the Kinect sensor provides a stable and accurate state estimation. The experimental results are also depicted in Fig. 6. The captured RGB image is shown in Fig. 6(a) and the top-view

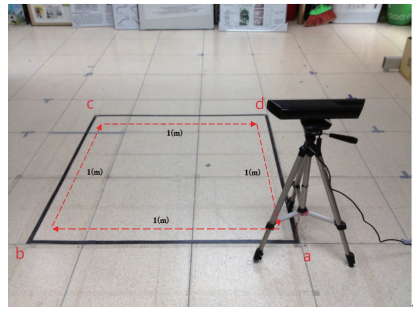
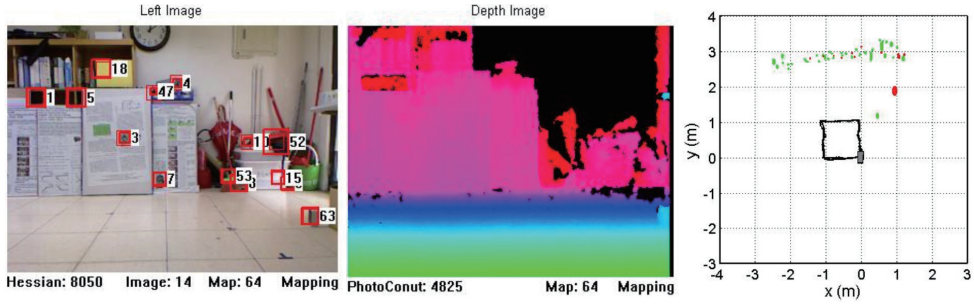


Fig. 5. (Color online) Trajectory of ground truth.

Table 3
Ground truth and estimated values.

Ground truth (m)			Estimated (m) (mean ± standard deviation)		
x	y	z	x ($\mu \pm \sigma$)	y ($\mu \pm \sigma$)	z ($\mu \pm \sigma$)
0.00	0.00	0.00	0.0029 ± 0.0032	0.0016 ± 0.0016	−0.0013 ± 0.0012
−1.00	0.00	0.00	−1.0565 ± 0.0307	−0.0402 ± 0.0019	0.0134 ± 0.0362
−1.00	1.00	0.00	−1.1165 ± 0.0315	1.00459 ± 0.0052	0.0211 ± 0.0031
0.00	1.00	0.00	0.0721 ± 0.0092	1.0364 ± 0.0036	0.0208 ± 0.0074



(a) (b) (c)

Fig. 6. (Color online) Images from 4825th frame.

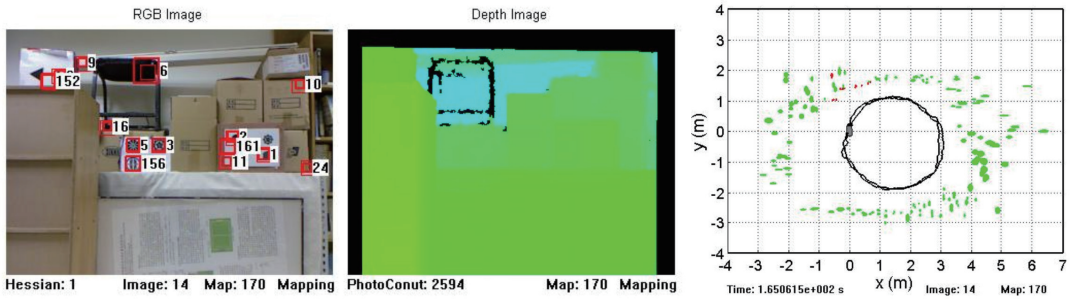
map is depicted in Fig. 6(c); the depth image is shown in Fig. 6(b). The (red) square marks in Fig. 6(a) indicate the stable landmarks extracted from the captured images. Fig. 6(c) shows a 2D plot of the estimated states of the camera and landmarks. The red (dark) ellipses represent the uncertainty of the observed landmarks, and the green (light) ellipses denote the unobserved landmarks. Note that the origin of the world coordinates in the map is defined as the initial position of the SLAM system.

5.2 *Robot SLAM*

In this experiment, the Kinect sensor is hand carried in a counterclockwise direction following a circular path of 3 m diameter. The camera lens is always frontward facing along the path. The SLAM system starts from the first image frame and captures image features with unknown positions. After these features are initialized, they are stored as landmarks in the map. As the Kinect sensor follows the circular path, the SLAM system concurrently builds the environmental map and estimates the Kinect pose. Figure 7 shows the final image frame obtained in the experiments where the captured RGB and depth images are shown in Figs. 7(a) and 7(b), respectively. The top-view plot of the environmental map is depicted in Fig. 7(c). Figure 8 depicts the deviations of the Kinect pose estimation along the *xyz*-axes. The figure shows that, during the SLAM task, the average pose deviation is less than 2 cm, and the highest peak is about 3 cm.

6. Conclusions

We developed an algorithm for robot simultaneous localization and mapping using a Kinect RGB-D sensor. In this study, we solved the misalignment problem of Kinect's multiple sensors as well as recovered the RGB image before the Kinect sensor could be applied to robot navigation. An image correction model was used to calibrate the RGB image, and a bilinear interpolation model was employed to align the RGB camera with the depth sensor. The novel calibration procedure reduced the RMSE from 16.05 to 2.18 pixels. In this study, we also extended the usability of SURF detectors in SLAM tasks



(a) (b) (c)

Fig. 7. (Color online) Images from last frame (2594th frame).

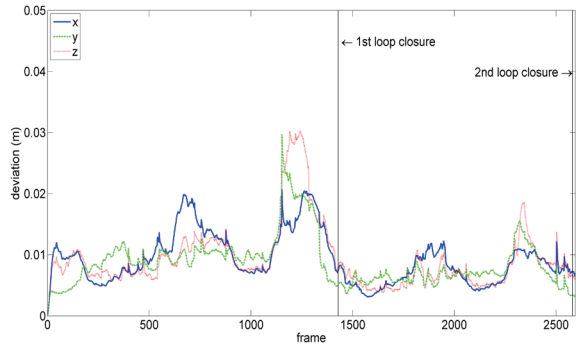


Fig. 8. (Color online) Deviations of Kinect pose along *xyz*-axes.

by utilizing their robust representation of visual landmarks. In the robot SLAM system, the SURF features were detected from the RGB images for building the environmental map. For each SURF feature, the RGB pixel coordinates were combined with the depth image to calculate the 3D coordinates. Two experiments were carried out to validate the performance of the RGB-D sensors for robot SLAM systems. The ground truth experiment demonstrated that the Kinect sensor can provide reliable and accurate measurement information for robot navigation. The robot SLAM experiment showed that the developed EKF-SLAM can deal with the loop-closure problem and correctly estimate the robot pose with a standard deviation of less than 2 cm.

Acknowledgements

This work was partially supported by the National Science Council of Taiwan under grant nos. NSC102-2221-E-032-045 and NSC101-2632-E-032-001-MY3.

References

1 Microsoft: KINECT for Windows, http://www.microsoft.com/en-us/kinectforwindows/ (accessed in April 2013).
2 Techbang: Body is controller, http://www.techbang.com/posts/2936-get-to-know-how-it-works- kinect (accessed in April 2013).
3 H. Yue, W. Chen, X. Wu and J. Liu: Opt. Laser Eng. **53** (2014) 104.
4 Z. Zhang: IEEE Trans. Pattern Anal. **22** (2000) 1330.
5 J. Heikkila and O. Silven: Proc. CVPR (IEEE, 1997) p. 1106.
6 J. Y. Bouguet: Camera Calibration Toolbox for Matlab, http://www.vision.caltech.edu/bouguetj/ calib_doc/ (accessed in April 2013).
7 A. Canessa, M. Chessa, A. Gibaldi, S. P. Sabatini and F. Solari: J. Visual Commun. Image Represent. **24** (2013) 1469.
8 T. Lindeberg: Int. J. Comput. Vision **30** (1998) 79.
9 H. Bay, A. Ess, T. Tuytelaars and L. V. Gool: Comput. Vision Image Understanding **100** (2008) 346.
10 G. Welch and G. Bishop: An Introduction to Kalman Filter (UNC-Chapel Hill TR 95-041, Chapel Hill, North Carolina, 2006).
11 R. C. Gonzalez and R. E. Woods: Digital Image Processing (Prentice Hall, Upper Saddle River, New Jersey 2007).
12 S. Hutchinson, G. D. Hager and P. I. Corke: IEEE Trans. Robot. **12** (1996) 651.
13 L. Sciavicco and B. Siciliano: Modeling and Control of Robot Manipulators (McGraw-Hill, New York, 1996).
14 A. J. Davison, I. D. Reid, N. D. Molton and O. Stasse: IEEE Trans. Pattern Anal. **29** (2007) 1052.
15 L. M. Paz, P. Pinies, J.D. Tardos and J. Neira: IEEE Trans. Robot. **24** (2008) 946.
16 P. A. Viola and M.J. Jones: Proc. CVPR (IEEE, 2001) p. 511.
17 G. Shakhnarovich, T. Darrell and P. Indyk: Nearest-Neighbor Methods in Learning and Vision (The MIT Press, Cambridge, MA, 2006).