American Journal of Signal Processing 2018, 8(1): 9-19
DOI: 10.5923/j.ajsp.20180801.02

Optimization of Transformation Matrix for 3D Cloud Mapping Using Sensor Fusion

Hai T. Nguyen*, Viet B. Ngo, Hai T. Quach

Department of Industrial Electronic-Biomedical Engineering, Faculty of Electrical-Electronics Engineering,
HCMC University of Technology and Education, Vietnam

Abstract This paper proposes an optimization method of transformation matrix for 3D cloud mapping for indoor mobile platform localization using fusion of a Kinect camera system and encoder sensors. In this research, RGB and depth images obtained from the Kinect system and encoder data are calculated to produce transformation matrices. A Kalman filter algorithm is applied to optimize these matrices and then produce a transformation matrix which minimizes cumulative error for building 3D cloud mapping. For mobile platform localization in an indoor environment, a SIFT algorithm is employed for feature detector and descriptor to determine similar points of two consecutive image frames for RGB-D transformation matrix. In addition, another transformation matrix is reconstructed from encoder data and it is combined with the RGB-D transformation matrix to produce the optimized transformation matrix using Kalman filter. This matrix allows to minimize cumulative error in building 3D point cloud image for robotic localization. Experimental results with mobile platform in door environment will show to illustrate the effectiveness of the proposed method.

Keywords Kinect camera, Transformation matrix, SIFT algorithm, Kalman filter, 3D point clouds, Encoder data

1. Introduction

An automatic mobile platform designed to automatically move based on 3D mapping has attracted researchers in recent years. In particular, robotic mapping is one of the most vital tasks in automatic robotic applications, in which the robot has to be supported the model of the navigational space in order to locate itself when moving. Thus, the map is essential for path planning processes in proposing the roadmap to target positions [1].

The robotic mapping is divided into 2D and 3D mappings. The 2D mapping has some disadvantages compared with 3D mapping. It means that the applications using sonar or laser sensors in navigation with 2D mapping has its drawbacks [2-4]. One of the great drawbacks of using the 2D mapping for accurate robotic locations is the lack of information in the third space dimension [5].

For improving the negative trends of 2D planning methods, the 3D mapping algorithms have been continuously developing with the supports of famous classical findings. Therefore, a Simultaneous Localization and Mapping (SLAM) algorithm is applied with 3D mapping methods for determination of the 3D model of large scale environments [6, 7]. The result is that the 3D mapping methods have effectively employed for building robotic mapping.

The quality of 2D mapping with obstacles from surroundings using sonar or laser sensors [8] is mainly determined due to its limitations. In particular, the sonar sensors used to obtain the ranges from the robot to surrounding obstacles just show obstacle information on the beam plane where the sensors are installed. For improvement of using the sonar sensor, the stereo camera system were used to take advantage of building 3D mapping with obstacles [8-11], but the calculation time to reconstruct the 3D ranging information from stereo images is expensive and it price is problem.

The Kinect RGB-D sensor (Kinect camera system) has been used to replace the stereo camera system for robotic localization [12]. This kind of the RGB-D sensor has not only the suitable accuracy, but also allow to calculate processing time with the fast speed. In addition, it is much cheaper than the 3D sensor with the same functions and easy to install for use. Therefore, algorithms have been applied using RGB-D sensors for determining the moving space as well as identification and positioning in the space of self-propelled robots in recent years [13-15]. One disadvantage of this RGB-D sensor is that its depth information is often noisy. Therefore, if a robot equipped with the Kinect moves long distances, the accumulated error for robotic localization gets over time [16]. There have been many proposed methods such as considering noise characteristics, dependently updating distances to reduce this error as well as to improve the accuracy of 3D mapping

* Corresponding author:
nthai@hcmute.edu.vn (Hai T. Nguyen)
Published online at http://journal.sapub.org/ajsp

[17-19].

In addition, for determination of feature-based image, the method of the feature-based image registration includes two parts: interest point detection and interest point description. Moreover, this method is based on the basis of Scale-Invariant Feature Transform (SIFT) algorithm for noisy reduction [20-22]. The features extracted also have scale and rotation-invariant performance with respect to illumination change, affine and perspective transformation. In addition to three aspects of repeatability, distinctiveness, and robustness, computation cost is not expensive.

The Kalman filter has many uses in applications of control, computer vision, filter and navigation [23-24]. In particular, the Kalman filter has been applied to track a vision object, in which it can be used to predict a process state. In addition, the Kalman filter was used in reconstructing medical images for contrast and transparency. In the robotic localization, the Kalman filter is employed to optimize the robotic position by processing the transformation matrices built from data obtained the Kinect camera system and the encoders.

In this paper, an optimization method by processing the transformation matrices built from sensor data for accurately reconstructing the 3D mapping. A Kinect (RGB-D) camera system is installed with the robot's cage to continuously capture the separate 2D image frames and 3D point clouds. All corresponding 2D points between the two consecutive image frames are estimated using the SIFT algorithm to produce the first transformation matrix. While the encoders equipped with the robotic wheels for calculating the second transformation matrix. The Kalman filter is applied to optimize them and create the most accurate transformation matrix. The next section of the paper presents in more details with the effectiveness of experimental illustrations.

2. Description of Mobile Platform

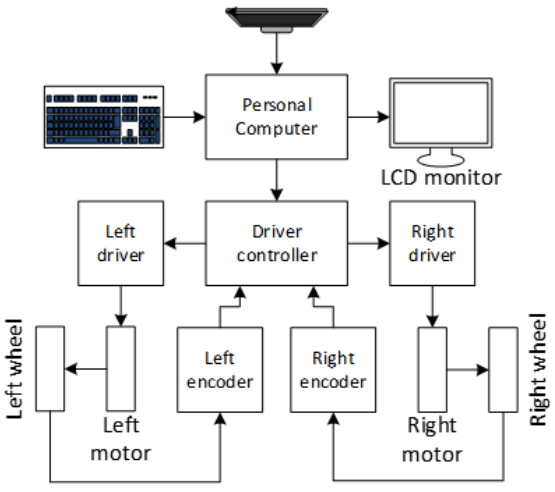

Figure 1. Block diagram of the hardware system of the mobile platform

In this research, the hardware system architecture of the mobile platform (robot) includes a Kinect RGB-D sensor V2 connected to a personal computer (PC) and other devices for processing data and controlling the robot. After navigation tasks for control of the differential robot, the velocity signals from the PC through Driver controller are sent to the left and right wheels. In addition, two encoders are installed with motors to send distance signals to the PC through Driver controller as shown in Figure 1. Finally, all mappings and localization processes are displayed on the LCD monitor during movement of the robot.

Figure 2 shows the robot model with the size of (400×350×365) mm. The height from the ground to the camera is 440 mm, each wheel's radius is 48 mm, the baseline between two wheels is 430 mm and two wheels are installed with two (12×64) pulses encoders.

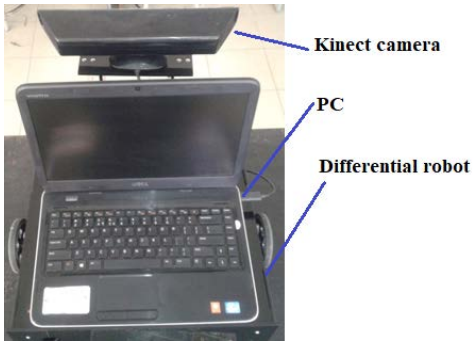

Figure 2. Robot model with the Kinect RGB-D sensor V2 and PC

3. Methods for Localizing and Mapping

The description of 3D mapping and methods for calculation of robotic localization are represented in the paper. The 3D mapping procedure shows conversion of 2D image into 3D image and solutions for image matching, synthesis, concatenation to create the optimized transformation matrix.

3.1. SIFT Algorithm for Detector and Descriptor of Features

For calculation of image features, the SIFT algorithm for feature detector and feature descriptor is employed [22]. The SIFT algorithm allows to detect features of 2D and 3D images, including two main steps: the first step is the feature detection and the second step is the feature description. In practice, the locations of stable features are detected, and then each feature is described so that it can be stable in various scale and direction appeared in the detecting images. Each keypoint when finishing description has the description of multiple direction vectors as shown in Figure 3.

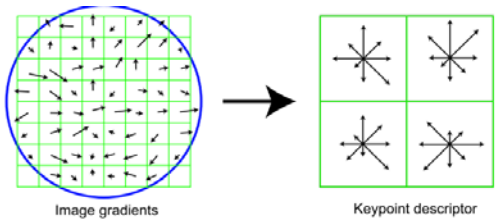

Figure 3. Image gradients and keypoint descriptor

In particular, two key points in image are locally matched together if the condition $d_{ij} < \varepsilon$ is satisfied and its equation is described as follows:

$$d_{ij} = \sqrt{(x_{i1} - x_{i1})^2 + (x_{i2} - x_{i2})^2 + \ldots + (x_{in} - x_{in})^2} \quad (1)$$

in which d_{ij} represents the distance between the i^{th} and j^{th} keypoints, and ε is the predefined matching condition.

In this research, the SIFT algorithm for feature detector and descriptor is used in describing and detecting the landmark. Each landmark is presented as a set of feature points, so it is detected when the total number of the detected feature points is larger than the predefined condition. The condition number in this project is unchanged throughout the landmarks, and it is chosen based on the experimental procedure.

3.2. Calculation of Transformation Matrix

For reduction of cumulative errors, transformation matrices built from data of the Kinect camera and encoders are considered. 2D images are calculated to convert into 3D images to create the matrices.

3.2.1. Conversion of 2D Image into 3D Image

A Kinect camera system with a main structure, consisting of a color camera, gives a 2D color image of 640x480 pixels. Each pixel will contain 3 RGB colors. In addition to the Kinect camera, the infrared light allows to captures the depth from the camera to the image plane as shown in Figure 4. Therefore, with many depth distances in the image, one can obtain the set of depths.

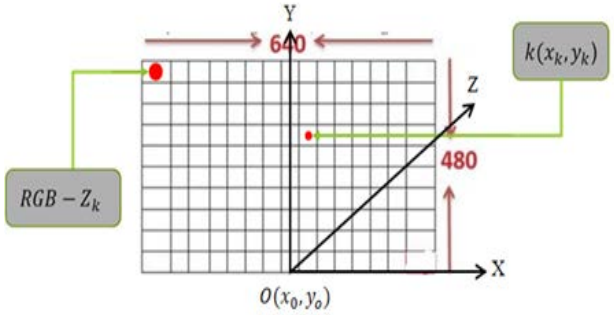

Figure 4. Description of the 3D image

To convert 2D pixel data $k(x_k, y_k)$ to 3D pixel data $K(X_k, Y_k, Z_k)$, we can use the following formulas:

$$X_k = \frac{z_k}{f}(x_k - x_o + \delta_x) \quad (2)$$

$$Y_k = \frac{z_k}{f}(y_k - y_o + \delta_y) \quad (3)$$

$$Z_k = Z_k \quad (4)$$

where (x_o, y_o) is the coordinate of the O origin of a 2D image, (δ_x, δ_y) is the distortion parameters of the lens obtained during adjusting the Kinect camera system.

3.2.2. SIFT Algorithm for Determination of Image Feature

After analyzing the features on two 2D images, a set of vectors describing the characteristics of the two 2D images are obtained. Therefore, the set of the first vectors with the features is compared to that of the second vectors for determining similar points. If the number of matching points satisfies the requirement, it means that two 2D images are similar (considered as one object captured at two different angles).

Thus, the process of finding pairs of similar features is carried out in three steps: finding the location of the feature point on two 2D images; describing the characteristics of each location found using the SIFT algorithm; and identifying pairs of similarities on the two images captured by the camera. After finding the similar point pairs in the two 2D images, one can determine the coordinates of the similar pairs in the two corresponding 3D clouds of the two 2D images.

From the coordinates of the similar points in the two 2D images, one can derive the coordinates of the corresponding point pairs on the 3D cloud based on (2), (3) and (4). In particular, the first 2D image gives a 3D cloud corresponding to the 3D coordinate ($O_0X_0Y_0Z_0$) and The second one has the corresponding coordinate ($O_1X_1Y_1Z_1$). Thus, after identifying the similar points with the 3D coordinates between the two 3D clouds, it can be paired these similar points.

3.2.3. Synthesis of Two 3D Point Clouds

Assume that there are two sets of 3D points which are $\boldsymbol{P}_0$ with the coordinate ($O_0X_0Y_0Z_0$) and $\boldsymbol{P}_1$ with the coordinate ($O_1X_1Y_1Z_1$) acquired from the RGB-D camera. Moreover, the rotation and translation matrix $\boldsymbol{E}$ is described as follows:

$$\boldsymbol{E} = \begin{bmatrix} e_{11} & e_{12} & e_{13} & e_{14} \\ e_{21} & e_{22} & e_{23} & e_{24} \\ e_{31} & e_{32} & e_{33} & e_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

in which the matrix of the point cloud $\boldsymbol{P}_1$ is calculated as follows:

$$\boldsymbol{P}_0 = \boldsymbol{P}_1 \times \boldsymbol{E} \quad (6)$$

$$\begin{bmatrix} x_{Mi} \\ y_{Mi} \\ z_{Mi} \\ 1 \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} & e_{13} & e_{14} \\ e_{21} & e_{22} & e_{23} & e_{24} \\ e_{31} & e_{32} & e_{33} & e_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x_{Ni} \\ y_{Ni} \\ z_{Ni} \\ 1 \end{bmatrix} \quad (7)$$

where $M_i(x_{Mi}, y_{Mi}, z_{Mi})$ is the i^{th} feature point of the cloud $\boldsymbol{P}_0$ and $N_j(x_{Nj}, y_{Nj}, z_{Nj})$ denotes the j^{th} feature point of the cloud $\boldsymbol{P}_1$. Assume that the points M_i and N_j with $(i = j)$ are the similar pairs. Moreover, 12 variances of the matrix $\boldsymbol{E}$ need to be determined. Therefore, in order to find these 12 variances, one needs to determine at least 4 pairs of similar points. It means that using Eq. (7) to solve 12 equations in the matrix $\boldsymbol{E}$, in which e_{14}, e_{24}, e_{34} are coordinates of the robot at this moment. Figure 5 describes two clouds with similar points

considered at two different coordinates.

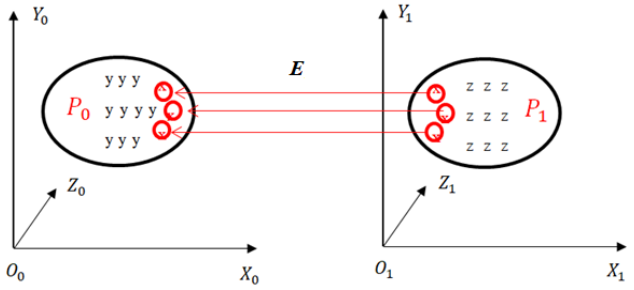

Figure 5. P_1 and P_0 are the transformation matrices

When performing the coordinate transformation ($O_1X_1Y_1Z_1$) into the coordinate ($O_0X_0Y_0Z_0$), the similar points x of the two clouds will be close to each other (almost identical). It means that the cloud P_1 approaches the cloud P_0 and it forms the larger cloud as shown in Figure 6.

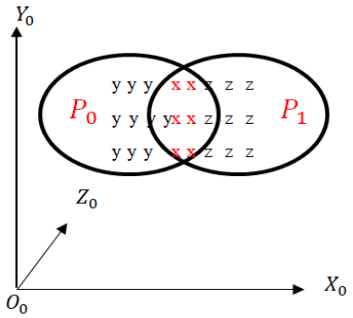

Figure 6. Two clouds have the similar point pairs x with red color

3.2.4. 3D Point Cloud Concatenation

All 3D point clouds are concatenated together based on the pair of the transformation matrices. Thus, for *(n+1)* clouds, the transformation matrix is calculated by the following formula:

$$\boldsymbol{E} = \boldsymbol{E}_{n(n-1)} \times \boldsymbol{E}_{(n-1)0} \tag{8}$$

where $\boldsymbol{E}$, $\boldsymbol{E}_{n(n-1)}$ and $\boldsymbol{E}_{(n-1)0}$ are the transformation matrices, which move the coordinates of the n^{th} cloud to the coordination system ($O_0X_0Y_0Z_0$). Therefore, the coordinates of the n^{th} cloud after transformed to the coordination system ($O_0X_0Y_0Z_0$) are represented as follows:

$$\boldsymbol{P}_n\left(O_0X_0Y_0Z_0\right) = \boldsymbol{P}_n\left(O_nX_nY_nZ_n\right) \times \boldsymbol{E} \tag{9}$$

In this case, the calculation of the transformation matrix using the recalculation method and it causes the cumulative position error of the robot.

3.2.5. Determination of the Transformation Matrix from Encoder Data

The dynamic equation for the robot describes the relationship between the coordinates $O(x,y)$ in the Descartes coordinate of the robot and the velocity of two robotic wheels. Figure 7 gives the model of a robot, in which the robot model will move and navigate by two wheels equipped with two encoders.

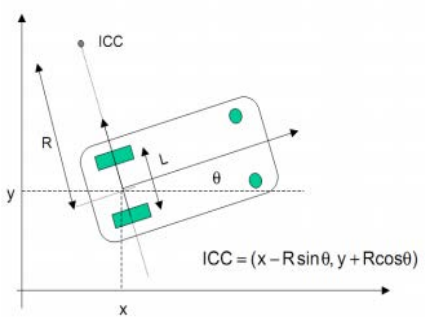

Figure 7. Model of the mobile platform

The robot will change direction based on changing the speed of the left wheel $v_l(t)$ and the right one $v_r(t)$. The Instantaneous Center of Curvature (ICC) is the instantaneous point that the robot will move in the trajectory of the curve around this point with the velocity $\omega(t)$. R is the distance from the ICC point to the midpoint of the two wheels. L is the distance between two wheels. $O(x,y)$ is the coordinate of the robot. θ is the angle of the robot chassis with the horizontal axis. The ICC has coordinates (x - $Rsin\theta$, y + $Rcos\theta$). Therefore, when the left wheel of the robot moves around the ICC point with the radius of an orbit (R - $L/2$), its right one will have the radius (R + $L/2$). Thus, the left and right wheels of the robot have the same angular velocity as the ICC and it is represented as follows:

$$\omega(t) = \frac{v_l(t)}{R - \frac{L}{2}} \tag{10}$$

$$\omega(t) = \frac{v_r(t)}{R + \frac{L}{2}} \tag{11}$$

From (10) and (11), one has:

$$\omega(t) = \frac{v_r(t) - v_l(t)}{L} \tag{12}$$

$$R = \frac{L}{2}\frac{v_r(t) + v_l(t)}{v_r(t) - v_l(t)} \tag{13}$$

Assume that $v(t)$ is the long velocity of the robot, one can calculate as follows:

$$v(t) = \omega(t) \cdot R = \frac{v_r(t) + v_l(t)}{2} \tag{14}$$

All these above components are considered at instantaneous time t, in which the coordinates ($x(t)$, $y(t)$) and the orientation angle $\theta(t)$ of the robot at time t are related to the angle velocity $\omega(t)$ and the long velocity $v(t)$, and it is calculated by the following equation:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \cos\theta(t) & 0 \\ \sin\theta(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} \tag{15}$$

From (12), (14) and (15), the dynamic equation of robot is determined as follows:

$$\begin{bmatrix}\dot{x}(t)\\ \dot{y}(t)\\ \dot{\theta}(t)\end{bmatrix}=\begin{bmatrix}\cos\theta(t) & 0\\ \sin\theta(t) & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\frac{v_r(t)+v_l(t)}{2}\\ \frac{v_r(t)-v_l(t)}{L}\end{bmatrix} \quad (16)$$

In the case of a 3D coordinate system, the camera is always installed at the constant height during moving, the equation is calculated at the coordinates *x* and *z* as follows:

$$\begin{bmatrix}\dot{x}(t)\\ \dot{z}(t)\\ \dot{\theta}(t)\end{bmatrix}=\begin{bmatrix}\cos\theta(t) & 0\\ \sin\theta(t) & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\frac{v_r(t)+v_l(t)}{2}\\ \frac{v_r(t)-v_l(t)}{L}\end{bmatrix} \quad (17)$$

Assume that two robotic wheels are installed with two encoders having 768 pulses, this means that when the wheel rotates a loop, the encoder gives 768 pulses. In this model, the wheel has the radius of R=47.5 mm, its circumference is C=2Rπ=298.3 mm, meaning that when the wheel moves about 298.3 mm, the wheel exactly rotates a loop and the formula can be written as follows:

$$\begin{bmatrix}x(n+1)\\ z(n+1)\\ \theta(n+1)\end{bmatrix}=\begin{bmatrix}\cos\theta(n+1) & 0\\ \sin\theta(n+1) & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\frac{d_r(n+1)+d_l(n+1)}{2}\\ \frac{d_r(n+1)-d_l(n+1)}{L}\end{bmatrix}+\begin{bmatrix}x(n)\\ z(n)\\ \theta(n)\end{bmatrix} \quad (18)$$

in which $d_r(n+1)$ and $d_l(n+1)$ are the distances of the right and left wheels, respectively moved from the n^{th} point to the $(n+1)^{th}$ point. In Eq. (18), it shows that the coordinates of the robot at time $(n+1)$ are determined based on the measured distance from movement of the left and right wheels from *n* to (*n*+1) and the coordinates of the robot at the moment *n*.

Therefore, reading the encoder pulse and the dynamic equation of the two robot wheels, one can know the position of the robot moving, as well as the coordinate of the point *O'*. Moreover, the rotation angle θ of the robot as well as the coordinate system (*O'X'Y'Z'*) around the axis *OY* are determined. Thus, the matrix $\boldsymbol{M}$ is determined as follows:

$$\boldsymbol{M}=\begin{bmatrix}\cos\theta(n+1) & 0 & \sin\theta(n+1) & x(n+1)\\ 0 & 1 & 0 & y\\ -\sin\theta(n+1) & 0 & \cos\theta(n+1) & z(n+1)\\ 0 & 0 & 0 & 1\end{bmatrix} \quad (19)$$

3.3. Optimization of Transformation Matrix Using Kalman Filter

In this project, the Kalman filter is applied to optimize the robot localization for optimizing the transformation matrix of two 3D clouds. This robot has two sets of sensors, in which the first sensor is a Kinect camera system that collects RGB data images with depth and the second one has two encoders installed with wheels for collecting data of the rotation wheels.

Assume that the Kinect data provide the estimated position q_1 of the robot at time *t*, data of the encoders show the estimated position q_2 of the robot at time $(t+1)$. These data exist Gaussian noises, called the combinational variances σ_{12} and σ_{22}. A least squares technique is applied to estimate the robot position, where $\hat{q}$ is the best estimated position of the robot and w_i is the weight of the i^{th} measurement, the estimated equation is described as follows:

$$S=\sum_{i=1}^{n} w_i(\hat{q}-q_i)^2 \quad (20)$$

To find the smallest error, one considers the derivative of *S* so that $\hat{q}$ equals to 0 and its equation is calculated as follows:

$$\frac{\partial S}{\partial \hat{q}}=\frac{\partial}{\partial \hat{q}}\sum_{i=1}^{n} w_i(\hat{q}-q_i)^2=2\sum_{i=1}^{n} w_i(\hat{q}-q_i)=0 \quad (21)$$

or

$$\sum_{i=1}^{n} w_i\hat{q}-\sum_{i=1}^{n} w_i q_i=0 \quad (22)$$

thus

$$\hat{q}=\frac{\sum_{i=1}^{n} w_i q_i}{\sum_{i=1}^{n} w_i} \quad (23)$$

where

$$w_i=\frac{1}{{\sigma_i}^2} \quad (24)$$

Substituting (24) into (23), with $n = 2$, one can rewrite as follows:

$$\hat{q}=\frac{{\sigma_2}^2}{{\sigma_1}^2+{\sigma_2}^2}q_1+\frac{{\sigma_1}^2}{{\sigma_1}^2+{\sigma_2}^2}q_2 \quad (25)$$

$$\frac{1}{\sigma^2}=\frac{1}{{\sigma_1}^2}+\frac{1}{{\sigma_2}^2} \quad (26)$$

$$\sigma^2=\frac{{\sigma_1}^2{\sigma_2}^2}{{\sigma_1}^2+{\sigma_1}^2} \quad (27)$$

It is obvious that σ^2 is always smaller than σ_1^2 and σ_2^2, this means that the best estimated position $\hat{q}$ is determined from two positions of two sets of the sensors.

From Eq. (25), the best localization equation is represented as follows:

$$\hat{q}=q_1+\frac{{\sigma_1}^2}{{\sigma_1}^2+{\sigma_2}^2}(q_2-q_1) \quad (28)$$

The method of determining the transformation matrix from the similar points gives the result of the high accuracy. This method can apply for calculating the precision matrix during the robot movement with short distances. In addition, the computation of the transformation matrix from the

similar points will minimize the cumulative error.

In building the transformation matrices, sensors install with the robot play an important role. In particular, two encoders of the two robot wheels provide pulses about speeds and distances during robot movement. The positioning block in the robot will convert the received data from the encoders into the robot coordinate with the *O* origin and the rotation angle of the robot around the *OY* axis. Therefore, data will be calculated to produce the transformation matrix ***M*** with fast time. While data of the camera Kinect will be calculated to produce the transformation matrix ***E***. The combination of two transformation matrices will create the optimized transformation matrix ***T***, which can improve processing speed.

$$\boldsymbol{T} = \begin{bmatrix} e_{11} & e_{12} & e_{13} & x_{\hat{q}} \\ e_{21} & e_{22} & e_{23} & y_{\hat{q}} \\ e_{31} & e_{32} & e_{33} & z_{\hat{q}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{29}$$

From Eq. (28) and Eq. (29), in which $\hat{q}\left(x_{\hat{q}}, y_{\hat{q}}, z_{\hat{q}}\right)$ is the most accurate coordinate of the robot and $q_1(x_{q1}, y_{q1}, z_{q1})$ is the estimation parameter obtained from the transformation matrix ***E***, $q_2(x_{q2}, y_{q2}, z_{q2})$ is the estimation obtained from the encoder, ${\sigma_1}^2$ and ${\sigma_2}^2$ are the variances representing the Gaussian signals for two positions q_1 and q_2. Thus, equations are determined as follows:

$$x_{\hat{q}} = x_{q_1} + \frac{{\sigma_1}^2}{{\sigma_1}^2 + {\sigma_2}^2}\left(x_{q_2} - x_{q_1}\right) \tag{30}$$

$$y_{\hat{q}} = y_{q_1} + \frac{{\sigma_1}^2}{{\sigma_1}^2 + {\sigma_2}^2}\left(y_{q_2} - y_{q_1}\right) \tag{31}$$

$$z_{\hat{q}} = z_{q_1} + \frac{{\sigma_1}^2}{{\sigma_1}^2 + {\sigma_2}^2}\left(z_{q_2} - z_{q_1}\right) \tag{32}$$

3.4. Point Cloud Transformation

The transformation method in 3D spaces is mainly used in this project due to its suitableness and effectiveness for the transformation of the 3D point cloud. In practice, a set of 3D point cloud is transformed to other positions in the same coordinate and its equation of a single 3D point is represented as follows:

$$\boldsymbol{P}_{i+1} = \boldsymbol{T} \times \boldsymbol{P}_i \tag{33}$$

in which $\boldsymbol{P}_i$ is the input point cloud, $\boldsymbol{P}_{i+1}$ is the ouput point cloud and ***T*** is the transformation matrix obtained from Eq. (29).

4. Results and Discussion

4.1. Image and Point Cloud Acquisition

The Kinect camera sensor (RGB-D camera) used in the paper produces image data comprising of RGB images with depth. Moreover, the depth image data is often pre-processed by the Kinect hardware as shown in Figure 8. In two consecutive images captured from the Kinect, point clouds contain both new and old information; consequently, the combination of them is expected to cover more additional data. The next step of concatenating process is to estimate the pixel locations of key points from the two images. Figure 9 shows two 3D clouds of one image frame at the room angle processed based on the RGB image data and depth information.

(a) (b)

Figure 8. Two consecutive RGB images captured from the Kinect sensor

Figure 9. 3D cloud images combined by RGB and depth images of the room space

4.2. Key Point Estimation

The SIFT algorithm was applied to locate the key points in the first and second images and their characteristics are independent from scales and rotations. Figure 10 shows the key points marked in the white points. The key points are mainly focused on the areas where the difference is in high gray level. Next, each detected key point is described by a 128-dimensions vector to be able to recognize easily by using Euclid's distance Eq. (1). The two key points are considered to be matched if the distance between their vectors is less than a pre-defined constant.

(a) (b)

Figure 10. Feature points of the first and second RGB images

4.3. 3D Key Point Matching

After estimated and described by using the SIFT algorithm, the key points on the second image are matched to their corresponding points and Figure 11 shows all corresponding key points on the first and second images. In this figure, each matching is presented by a green line connected between two key points.

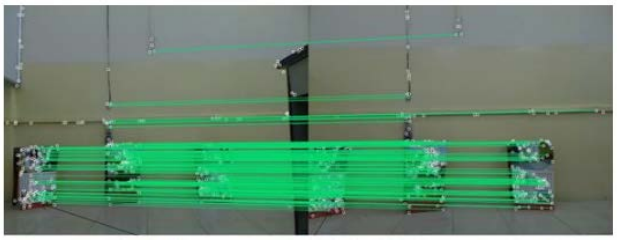

Figure 11. Pairs of similar points of two consecutive RGB images connected in green color

Figure 12 shows matching red lines which are projected to the 3D space.

Figure 12. Matching keypoints between the first and second 3D point clouds

4.4. Pair Concatenation

In Eq. (5), the transformation matrix has the size of (4×4), in which 12 parameters are unknown. Therefore, it has at least 12 pairs of corresponding points in the 3D cloud in order to infer the correct values. However, the number of corresponding points is often more than 12 due to the errors can happen when matching between two point clouds. After the transformation matrix is determined, the matrix is applied to calculate for the second point cloud in order to have the same coordinate system as the first point cloud. The experimental result of the concatenating point cloud is described in the Figure 13.

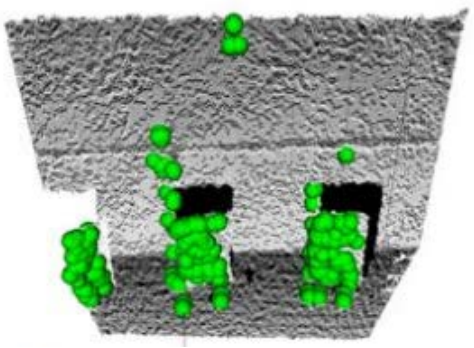

Figure 13. The concatenated point cloud

4.5. Optimization of 3D Mapping Using Kalman Filter

4.5.1. 3D Clouds with Less Similar Points in the Transformation Matrix

Figure 14 is a 3D map of the robotic path in the room environment without the Kalman filter. It is obvious that the rotating robot parts of the map occur during the grafting process, in which the right wall was broken and the left part of the room was misplaced. The cause of this error is that the number of similar points of two consecutive clouds at the location of the error is not sufficient to compute the transformation matrix. Therefore, the value of the matrix element is defined as zero corresponding to pairs of two clouds not being close together. The error value of this transformation matrix will affect all other transformation matrices in calculating.

a. 2D image of the room angle

b. 3D mapping after combining point clouds

Figure 14. 3D mapping with the robot path when moving around the room without Kalman filter

Figure 15 describes the 3D point cloud concatenation at the room space. We can see the quality of the initial grafting of the clouds very well, describing the relatively straight space of the room. But when near the corner of the room, there is a difference. The wall of the room was not straight and was broken. With the algorithms presented in the previous section, we can see that clustering of clouds in small numbers gives us very good results, but when the number of clouds increases, errors occur. The cause of this deviation lies in calculating the transformation matrix to multiply the clouds together.

Figure 16 is the result of clustering the clouds after optimizing the data using the Kalman filter. We can see that the location of the error due to the cumulative error has been corrected. The wall of the room was straight and the map of the room was more accurate. The positional signals measured from the encoder can be calculated via the kinetic equations as described in the next section.

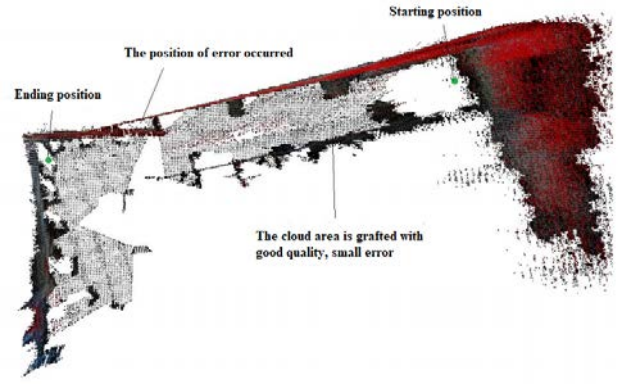

Figure 15. 3D map of a part of the room is faulty due to cumulative error

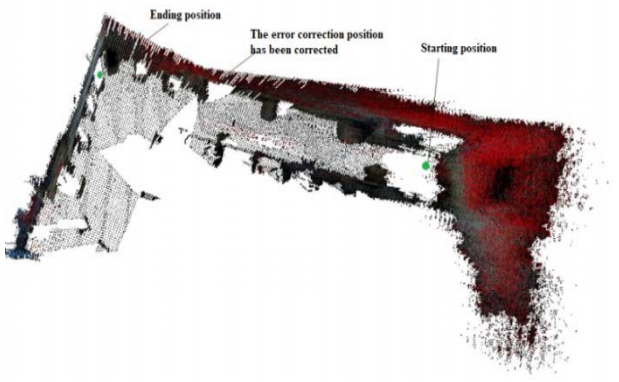

Figure 16. 3D map of a part of the room after eliminating the cumulative error with Kalman filter

4.5.2. Improvement of the Cumulative Error for Calculating the Transformation Matrix

In addition, the Kalman filter improves the cumulative error in the transformation matrix calculation. Figure 17 shows the X coordinate of the robot moving a straight distance of 30cm toward the front of the Kinect camera. With 21 RGB images and depth images, we can identify 21 different coordinates of the robot to different positions. The brown line is the standard coordinates of the robot when traveling in a straight line defined in advance.

When the robot moves straight in front of the Kinect camera in the direction of the Z axis, its X coordinate is zero at all positions. The green line is the value of the X coordinate of the robot measured from the encoders and it is calculated from the coordinates of the similar points. It means that the blue line tends to be far from the brown line due to the cumulative error. While the purple line determined using the Kalman filter is optimized more than the blue and green lines, so it is closer to the brown line. The demonstration result of the robot position error shows that it has been minimized using the Kalman filter and the encoder signals.

In similarity, the graphs with the Y coordinates of the robots at other positions were calculated, in which the blue line is determined the coordinate from the similar points, the green line is calculated based on the encoder signals and the purple one is determined using the Kalman filter as shown in Figure 18. Because the robot moves on the flat surface of the room, its height during moving is constant and the Y coordinate value of the robot is zero at the measuring locations. While the purple line is closest to the brown line and it is the best position of the robot.

In Figure 19, the Z-coordinate statistics of the robot are transmitted straight through 21 positions in the direction of the Z axis, two consecutive position are set to be 30cm, so the Z coordinate value of the robot is statistically continuous as the brown line. In addition, the blue line is far from the brown line compared to the purple line. From three graphs of Figs. 17, 18 and 19, the Kalman filter applied in this research minimizes the cumulative error when calculating the transformation matrices between 3D clouds.

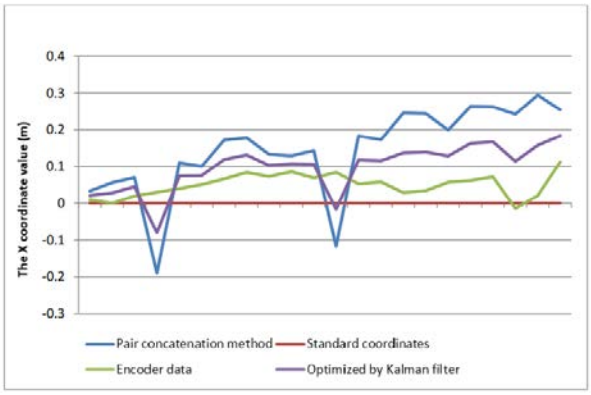

Figure 17. Statistic of the coordinate values X of the robot

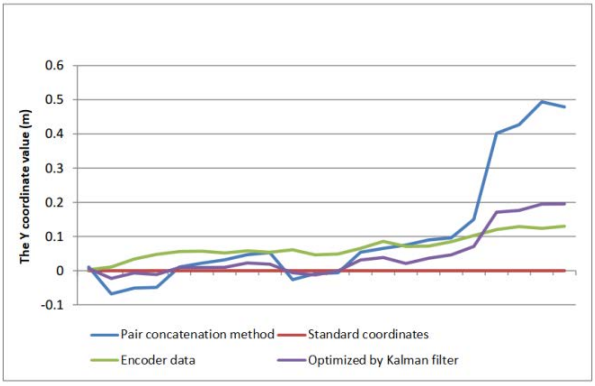

Figure 18. Statistic of the coordinate values Y of the robot

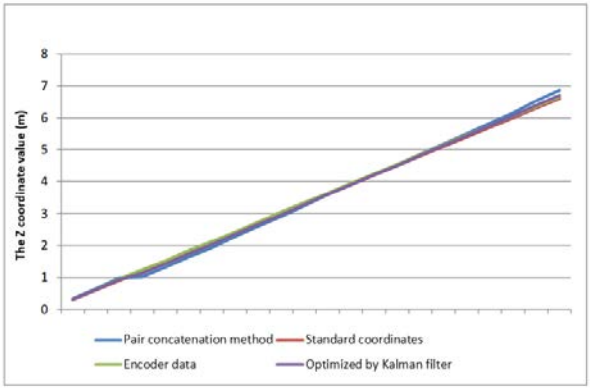

Figure 19. Statistic of the coordinate values Z of the robot

4.5.3. Experimental Results of Improved 3D Clouds at Different Room Angles during Robotic Movements

Figure 20 describes 2D images captured on the robotic pathway in the room environment. The robot moves along the paths to the left and to the right of the room. Therefore, these 2D images of the environment are converted into 3D clouds and then they are grafted to create the 3D spatial images as shown Figure 21, Figure 22 and Figure 23, which show three 3D cloud maps with the paths at different angles.

3D cloud mappings with the high accuracy at different room angles are shown using the optimized transformation matrix when the robot moves around.

(a) (b)

(c)

Figure 20. 2D maps during robotic movements for 3D loud mapping: (a)-Robot moving straight at the room left; (b)- Robot moving straight at the room right; (c)-A room angle

Figure 21. 3D map of the robot path with the first angle using Kalman filter

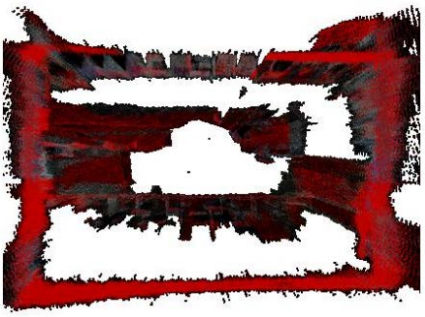

Figure 22. 3D map of the robot path with the second angle using Kalman filter

Figure 23. 3D map of the robot path with the third angle using Kalman filter

In recent years, 3D maps in indoor environment were built from the RGB-D data of the Kinect sensor for mobile vehicles. In order to build the 3D maps with the proposed RGB-D SLAM algorithm, data (Depth image and RGB image) from the Kinect camera are processed to produce RE-RANSAC-ICP and 8point-RANSAC [25]. In addition, some methods for feature extraction were applied and then matching among features was performed [26]. In particular, a new cloud-coupled optimization algorithm for presenting a full 3D mapping was worked out by combining image-based and shape-based alignment. Moreover, image information and its depth was combined to detect similarities of frames and then it was optimized to achieve consistent maps. To align the current frame and the previous frame, the alignment step was applied using the Iterative Closest Points (ICP) RGB-D algorithm based on the combination between RGB and depth information. It means that a detection of the similar points was performed using discrete points for matching the current frame and the previous frame. If a similar pair is detected, it is added to the model graph and a combining process is performed. After this alignment step, a new frame is added to the 3D model [15]. Furthermore, due to the accuracy related to the initial position of the features in the ICP algorithm, the RANSAC algorithm was utilized for optimizing the initial position of the features and then the exception points can be eliminated [18].

Methods of the Iterative Closest Points (ICP) and Random Sample Consensus (RANSAC) have been applied to optimize the initial position of features and then to remove the peripheral points based on the combination between RGB and depth information in recent years. It is obvious that these methods have the advantages of the simplicity, fast calculation and increased accuracy during a full 3D mapping. In addition, the Loop Closure technique to match similarities between two consecutive frames was employed for optimization of building 3D map and for merging point clouds between frames [15, 18, 25]. However, it has some problems related to accuracy of making 3D mapping. In practice, the combination of typical visual features is more accurate than that of thick-spot cloud, but this makes misleading in areas of image due to lacking visual information, such as very dark rooms. Moreover, the RGB-D maps only use two successive frames for estimating camera movements and the Loop Closure algorithm is applied for matching image features between frames, but it is not enough to create full 3D maps.

In order to improve exactly 3D map using the Kinect camera, in this paper, the proposed method is to optimize the transformation matrix which combined between the RGB data-based transformation matrix and the encoder data-based one using Kalman filter. In addition, the SIFT was employed for feature detector and feature descriptor. With this method, this transformation matrix will minimize the cumulative error in building 3D point cloud based on multiple consecutive cloud image frames. From this 3D point cloud, the coordinates of the robot are most accurately determined for the robotic localization. In addition to the proposed method for the optimized transformation matrix, the RGB-D camera sensor used in this research is much cheaper than other 3D stereo camera which was used to build 3D point cloud mapping for the robotic localization. of the same type. Therefore, the algorithm of combining the RGB-D camera and encoder sensors to determine the movement space that can be widely used in the field of identification and localization of mobile platform.

5. Conclusions

In the paper, the model of mobile platform (robot) in the indoor environment was represented and 3D point clouds were completely reconstructed based on RGB-D image frames obtained from the Kinect camera system. A SIFT algorithm was employed to detect and to describe image features. In addition, encoder data were used to combine to RGB image data and the Kalman filter was utilized to produce the optimized transformation matrix for minimize the cumulative error for robot localization. Experimental results prove that the effectiveness of combining between the Kinect camera system and the encoder sensors on the robot. Moreover, the Kinect can be cheaper cost computation compared to other stereo cameras for indoor applications.

ACKNOWLEDGEMENTS

This work is supported by HoChiMinh City University of Technology and Education (HCMUTE) under Grant T2017-60TD. We would like to thank HCMUTE, students and colleagues for supports on this project.

REFERENCES

[1] J. Fuentes-Pacheco, “Visual simultaneous localization and mapping: A survey,” Springer Science - Business Media Dordrecht, ed, pp. 55–81, 2015. ISSN: 1573-7462. DOI: 10.1007/S10462-012-9365-8.

[2] A. Basu, S. K. Ghosh and S. Sarkar, “Autonomous navigation and 2D mapping using SONAR,” in The 5th International Conference on Wireless Networks and Embedded Systems (WECON), Rajpura, 2016, pp. 1-5. ISBN: 978-1 5090-0893-3. DOI: 10.1109/WECON.2016.7993421.

[3] C. P. O. Diaz and A. J. Alvares, “A 2D-mapping strategy based on line segment extraction,” in 2010 IEEE ANDESCO N, Bogota, 2010, pp. 1-6. ISBN: 978-1-4244-6742-6. DOI: 10.1109/ANDESCON.2010.5633272.

[4] H. Iikura, S. Ogawa, K. Kobayashi and K. Watanabe, “Real-time 2D map building for an unmanned vehicle in a closed area,” in SICE 2003 Annual Conference (IEEE Cat. No.03TH8734), Fukui, Japan, 2003, pp. 1081-1085 Vol.1. ISBN: 0-7803-8352-4.

[5] R. Z. M. Dr. Wael R. Abdulmajeed, “Comparison Between 2D and 3D Mapping for Indoor Environments,” International Journal of Engineering Research and Technology, vol. 2, ed, 2013. ISSN 2278 – 0181.

[6] X. Liu, B. Guo and C. Meng, “A method of simultaneous location and mapping based on RGB-D cameras,” in The 14th International Conference on Control, Automation, Robotics and Vision (ICARCV), Phuket, 2016, pp. 1-5. ISBN: 978-1-5090-3549-6. DOI: 10.1109/ ICARCV.2016.7838786.

[7] E. Fernández-Moral, V. Arévalo and J. González-Jiménez, “Extrinsic calibration of a set of 2D laser rangefinders,” in 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, 2015, pp. 2098-2104. ISBN: 978-1-4799-6923-4. DOI: 10.1109/ICRA.2015.7139475.

[8] Thomas Whelan, Michael Kaess, Hordur Johannsson, Maurice Fallon, John J. Leonard, John McDonald, “Real-time large-scale dense RGB-D SLAM with volumetric fusion,” The International Journal of Robotics Research, Vol 34, Issue 4-5, pp. 598 – 626, December 9, 2014.

[9] Nguyen Thanh Hai, N T Hung, “A Bayesian Recursive Algorithm for Freespace Estimation Using a Stereoscopic Camera System in an Autonomous Wheelchair,” American J. of Biomedical Eng, Vol. 1, No. 1, pp. 44-54, 2011. -ISSN: 2163-1077. DOI: 10.5923/J.AJBE.20110101.08.

[10] Nguyen Thanh Hai, V T Kiet, “Freespace Estimation in an Autonomous Wheelchair Using a Stereoscopic Cameras System,” in The 32rd IEEE Annual International Conference on EMBS, 2010. ISBN: 978-1-4244-4123-5. DOI: 10.1109/IEMBS.2010.5626118.

[11] N B Viet, N T Hai, N V Hung, “Tracking Landmarks for Control of an Electric Wheelchair Using a Stereoscopic Camera System,” in The Inter. Conf. on Advanced Tech for Communications, pp. 12-17, 2013. ISBN: 978-1-4799-1089-2. DOI: 10.1109/ATC.2013.6698133.

[12] D. Schleicher, L. M. Bergasa, R. Barea, E. Lopez and M. Ocana, “Real-Time Simultaneous Localization and Mapping using a Wide-Angle Stereo Camera and Adaptive Patches,” in 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, 2006, pp. 2090-2095. ISBN: 1-4244-0258-1. DOI: 10.1109/IROS.2006.282486.

[13] C. Lim Chee, S. N. Basah, S. Yaacob, M. Y. Din, and Y. E. Juan, “Accuracy and reliability of optimum distance for high performance Kinect Sensor,” in Biomedical Engineering (ICoBE), 2015 2nd International Conference on, ed, pp. 1-7, 2015. ISBN: 978-1-4799-1749-5.
DOI: 10.1109/ICoBE.2015.7235927.

[14] G. Loianno, V. Lippiello and B. SicilianO, “Fast localization and 3D mapping using an RGB-D sensor,” in 2013 16th International Conference on Advanced Robotics (ICAR), Montevideo, 2013, pp. 1-6. ISBN: 978-1-4799-2722-7.
DOI: 10.1109/ICAR.2013.6766558.

[15] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, “Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments,” The International Journal of Robotics Research, vol. 31, no. 5, pp. 647–663, 2012. ISSN: 02783649. DOI: 10.1177/0278364911434148.

[16] T. Yamaguchi, T. Emaru, Y. Kobayashi and A. A. Ravankar, “3D map-building from RGB-D data considering noise characteristics of Kinect,” in 2016 IEEE/SICE International Symposium on System Integration (SII), Sapporo, 2016, pp. 379-384. ISBN: 978-1-5090-3329-4.
DOI: 10.1109/SII.2016.7844028.

[17] H. Jo, S. Jo, H. M. Cho and E. Kim, “Efficient 3D mapping with RGB-D camera based on distance dependent update,” in 2016 16th International Conference on Control, Automation and Systems (ICCAS), Gyeongju, 2016, pp. 873-875. ISBN: 978-89-93215-11-3. DOI: 10.1109/ICCAS.2016.7832417.

[18] B. Yuan and Y. Zhang, “A 3D Map Reconstruction Algorithm in Indoor Environment Based on RGB-D Information,” in 15th International Symposium on Parallel and Distributed Computing (ISPDC), Fuzhou, 2016, pp. 358-363. ISBN: 978-1-5090-4152-7.
DOI: 10.1109/ISPDC.2016.59.

[19] K. Lee, “Accurate Continuous Sweeping Framework in Indoor Spaces with Backpack Sensor System for Applications to 3D Mapping,” The IEEE Robotics and Automation Letters, vol. 1, no. 1, pp. 316-323, Jan. 2016. ISSN: 2377-3766.
DOI: 10.1109/LRA.2016.2516585.

[20] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” International Journal of Computer Vision, vol. 60, ed, pp. 91-110, 2004. ISSN: 1573-1405. DOI: 10.1023/B:VISI.0000029664.99615.94.

[21] Muthukrishnan, R1 and Ravi, J, “Image type-based Assessment of SIFT and FAST Algorithms,” International Journal of Signal Processing, Image Processing and Pattern Recognition, Vol.8, No.3 (2015), pp.211-216. ISSN: 2005-4254. DOI: 10.14257/IJSIP.2015.8.3.19.

[22] Nirvair Neeru and Lakhwinder Kaur, “Modified SIFT Descriptors for Face Recognition under Different Emotions,” Hindawi Publishing Corporation Journal of Engineering, Volume 2016, Article ID 9387545, 12 pages 1-12. ISSN: 2314-4912. DOI: 10.1155/2016/9387545.

[23] Radek Baranek, Frantisek Solc, “Model-Based Attitude Estimation for Multicopters,” Advances in Electrical and Electronic Engineering, volume: 12, number: 5, 2014 December, pp. 501-510. ISSN 1804-3119.
DOI: 10.15598/AEEE.V12I5.1151.

[24] Pavel Brandstetter, Marek Dobrovsky1, “Speed Estimation of Induction Motor Using Model Reference Adaptive System with Kalman Filter,” Advances in Electrical and Electronic Engineering, volume: 12, number: 1, 2013, pp. 22-28. ISSN 1804-3119. DOI: 10.15598/AEEE.V11I1.802.

[25] G. Hu, S. Huang, L. Zhao, A. Alempijevic and G. Dissanayake, "A robust RGB-D SLAM algorithm," 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, 2012, pp. 1714-1719.

[26] Huang A.S. et al, “Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera,” in Christensen H., Khatib O. (eds) Robotics Research. Springer Tracts in Advanced Robotics, vol. 100. Springer, Cham, 2017.