Proceedings of 2019 National Symposium on System Science and Engineering
National Taiwan Normal University, Taipei City, 2-4 April, 2019 國立臺灣師範大學

Autonomous Navigation of an Indoor Mecanum-Wheeled Omnidirectional Robot Using Segnet

Po-An Wei, Ching-Chih Tsai, Feng-Chun Tai
Department of Electrical Engineering, National Chung Hsing University, Taichung, Taiwan
Email: g106064501@mail.nchu.edu.tw, cctsai@nchu.edu.tw, d099064008@mail.nchu.edu.tw

***Abstract-*This paper proposes an autonomous navigation method using Segnet, a deep convolutional encoder-decoder architecture, for autonomous navigation of an indoor Mecanum-wheeled omnidirectional robot (MWOR). The autonomous navigation system is equipped with one Jetson TX2 module from Nvidia, one Laser scanner (LiDAR), one iBeacon location module, one webcam, and one wheeled omnidirectional mobile robot. The iBeacon location system is used to achieve find roughly global position estimate of the robot, and the laser scanner is employed to avoid any collisions from any static, or dynamic, or unexpected moving objects. A SegNet model using deep learning and TX2 module is utilized to proceed with image recognition acquired from the webcam, in order to generate motion commands to steer the robot autonomously. Experimental results are conducted to show the effectiveness of the proposed autonomous navigation method.**

***Keywords**: Deep learning, Omnidirectional mobile robot, position estimate, autonomous navigation, SegNet.*

I. INTRODUCTION

Deep learning is very popular in recent years. Deep learning models have sometimes achieved increasing success due to the availability of massive datasets and extenting model depth and parameterisation. Nevertheless, practical factors licluding memory and computational time during training and testing are important factors to consider while choosing a model from a large bank of models. Hence,the time of training turns into a major consideration particularly while the performance gain is not commensurate with increased training time as shown in our experiments.Test time memory and computational load are important to deploy models on specialised embedded devices.From an overall efficiency viewpoint, less attention has been paid to smaller and more memory, time efficient models for immediate applications such as road scene understanding.It was the primary motivation behind the proposal of SegNet, which is significantly than other competing architectures. SegNet has been shown efficient for tasks such as road scene understanding.

Autonomous navigation of mobile robots or vehicles is expected to provide various services within living environments of humans [1]. Such a robotic technology has been ready to show its practical use in industry, but, so far, the robots for industry simply follow a given motion by humans.Therefore, we will conduct as a means to allow elderly and physically impaired people to travel to destinations within public facilities such as airports. Nowadays, image segmentation is more popular such that it can be used to identify the regions of interest in a scene or annotate the data [2]. In image recognition by deep learning, areas within the image are recognized and classified as, for example, a person, road, sidewalk, or building. With this recognition method, the vehicle can estimate its position and direction of travel with great reliability. The authors in [3] used deep learning to achieve image recognition and extraction of the travelable area by proceessing the images acquired from monocular camera images mounted on autonomous cars. The authors in [4] showed that an electric mobile robot autonomously travels through a passage, and its own position and direction were estimated using deep learning.In particular, this system in [4] operated by using a camera in a smart phone to obtain an input image in order to make it as simple as possible.

Mecanum-wheeled omnidirectional robots (MWORs) have been widely used for our living life and industrial material handling, such as omnidirectional wheelchairs, automatic guided vehicles, and etc. There are two kinds of MWORs built by using 45-degree and 90-degree Mecanum wheels. Unlike conventional differential driving, MWORs have the superior flexibility to move towards any position and orientation. MWORs can be made using different wheel configurations including three wheels, four wheels, car-like four wheels, and etc.

Motivated by [1-4], this paper aims to develop and verify a self-driving system for an indoor car-like Mecanum-wheeled omnidirectional mobile robot that have varous applications in industry and our daliy life. The proposed techniques would provide references for professionals working in this area, especially for researchers and engineers working for personal care robots.

The rest of the paper is outlined as follows. Section II briefly introduces the system structure, physical configuration and kinematic model of the experimental MWOR. Section III delineates trajectory tracking of the experimental MWOR. Section IV describes the iBeacon modules and its triangulation method via Kalman filtering. Section V intrioduces the used Segnet model from deep learning, and then proposes the method to estimate the travaling area and its centerline. Experimental results are presented and discussed in Section VI. Section VII concludes the paper.

II.SYSTEM DESCRIPTION OF THE EXPERIMENTAL MWOR

2.1 System Structure

Fig. 1 shows the system structure of the experimental Mecanum-wheeled omnidirectional mobile robot (MWOR). Having acquired environmental images from the webcam module, a Jetson TX2 computing module determines the travelable area (floor surface) and uses it to generate a centerline and take control actions. The iBeacon module using Bluetooth is employed to find the current position of the MWOR, and the LiDAR (Light Detection and Ranging) device is exploited to achieve obstacle avoidance. To drive the MWOR to go along the determined centerline and prevent

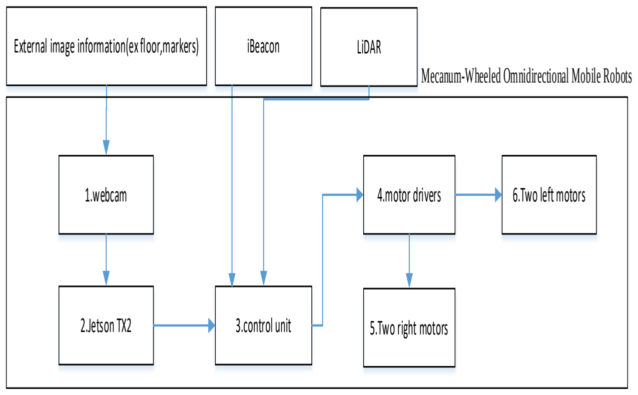
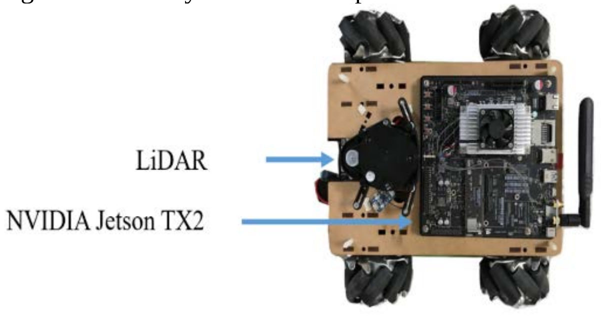


Fig. 1. System structure of the proposed autonomous navigation control system for the experimental MWOR.



(a)

(b) .

Fig. 2. Physical configuration of the experimental MWOR: (a) illustration of the used LiDAR and Nvidia Jetson TX2; (b) illustration of the used iBeacon module.

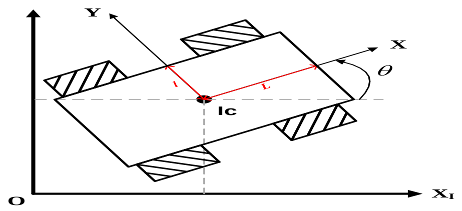


Fig. 3. Pose definition of the laboratory-built MWOR.

the MWOR from collisions, the MWOR controller thus calculates the rotation speed commands of its left and right motors via the proposed kinematic motion control law and obstacle avoidance approach, and then sends these commands to the four corresponding drivers. Fig. 2 displays the physical configuration of the experimental MWOR.

2.2 Kinematic Model of the Experimental MWOR

The subsection will briefly describe the Next, both forward and inverse kinematic models of the experimental MWOR are formulated in the global frame, where the pose vector $\boldsymbol{x}=\begin{bmatrix}x & y & \theta\end{bmatrix}^T$ denotes the position and orientation of the robot in the world frame as shown in Fig. 3. The forward kinematics of the experimental mobile robot is then described by

$$\begin{bmatrix}\dot{x}_w\\ \dot{y}_w\\ \dot{\theta}\end{bmatrix}=\boldsymbol{J}(\theta)\boldsymbol{\upsilon}_w \tag{1}$$

where $\boldsymbol{\upsilon}_w=[\upsilon_{1w}\ \upsilon_{2w}\ \upsilon_{3w}\ \upsilon_{4w}]^T$ represents the velocity vector of the four wheels and $\upsilon_{iw}, i=1,\ldots,4$, denotes the speed of the i-th wheel. Moreover, $\boldsymbol{J}(\theta)$ is given by

$$J(\theta)=\frac{1}{4}\begin{bmatrix}\sqrt{2}\sin(\theta_1) & \sqrt{2}\cos(\theta_1) & \sqrt{2}\cos(\theta_1) & \sqrt{2}\sin(\theta_1)\\ -\sqrt{2}\cos(\theta_1) & \sqrt{2}\sin(\theta_1) & \sqrt{2}\sin(\theta_1) & -\sqrt{2}\cos(\theta_1)\\ -\frac{1}{L+l} & \frac{1}{L+l} & -\frac{1}{L+l} & \frac{1}{L+l}\end{bmatrix} \tag{2}$$

where $\theta_1=\theta+\pi/4$, L and l respectively denote the length shown in Fig. 3. By using the pseudo inverse matrix, $\mathbf{J}^+$, of the matrix $\mathbf{J}$ where $\boldsymbol{J}\boldsymbol{J}^+=\mathrm{I}_3$, one expresses the inverse kinematics model of the robot by

$$\boldsymbol{\upsilon}_w=\boldsymbol{J}^+(\theta)\begin{bmatrix}\dot{x}_w\\ \dot{y}_w\\ \dot{\theta}\end{bmatrix} \tag{3}$$

where

$$\boldsymbol{J}^+(\theta)=\begin{bmatrix}\sqrt{2}\sin(\theta_1) & -\sqrt{2}\cos(\theta_1) & -(l+L)\\ \sqrt{2}\cos(\theta_1) & \sqrt{2}\sin(\theta_1) & (l+L)\\ \sqrt{2}\cos(\theta_1) & \sqrt{2}\sin(\theta_1) & -(l+L)\\ \sqrt{2}\sin(\theta_1) & -\sqrt{2}\cos(\theta_1) & (l+L)\end{bmatrix}$$

2.3 Kinematic Motion Control

This subsection will recall the kinematic control method of the MWOR for tracking any smooth differentiable trajectory $\begin{bmatrix}x_d(t) & y_d(t) & \theta_d(t)\end{bmatrix}^T\in C^1$. To this end, define the following tracking error vector

$$\begin{bmatrix}x_e(t)\\ y_e(t)\\ \theta_e(t)\end{bmatrix}=\begin{bmatrix}x_w(t)\\ y_w(t)\\ \theta(t)\end{bmatrix}-\begin{bmatrix}x_d(t)\\ y_d(t)\\ \theta_d(t)\end{bmatrix} \tag{4}$$

Taking the time derivative of (4) and using (2) yield obtains

$$\begin{bmatrix}\dot{x}_e(t)\\ \dot{y}_e(t)\\ \dot{\theta}_e(t)\end{bmatrix}=\begin{bmatrix}\dot{x}(t)\\ \dot{y}(t)\\ \dot{\theta}(t)\end{bmatrix}-\begin{bmatrix}\dot{x}_d(t)\\ \dot{y}_d(t)\\ \dot{\theta}_d(t)\end{bmatrix}=\boldsymbol{J}(\theta(t))\begin{bmatrix}r\omega_1(t)\\ r\omega_2(t)\\ r\omega_3(t)\\ r\omega_4(t)\end{bmatrix}-\begin{bmatrix}\dot{x}_d(t)\\ \dot{y}_d(t)\\ \dot{\theta}_d(t)\end{bmatrix} \tag{5}$$

Hence, the kinematic motion control law is proposed as below.

$$\begin{bmatrix}\omega_1(t)\\ \omega_2(t)\\ \omega_3(t)\\ \omega_4(t)\end{bmatrix}=\frac{1}{r}\boldsymbol{J}^+(\theta(t))\left(-\boldsymbol{K}_p\begin{bmatrix}x_e(t)\\ y_e(t)\\ \theta_e(t)\end{bmatrix}-\boldsymbol{K}_I\begin{bmatrix}\int_0^t x_e(\tau)d\tau\\ \int_0^t y_e(\tau)d\tau\\ \int_0^t \theta_e(\tau)d\tau\end{bmatrix}+\begin{bmatrix}\dot{x}_d(t)\\ \dot{y}_d(t)\\ \dot{\theta}_d(t)\end{bmatrix}\right) \tag{6}$$

where the two gain matrices, K_p and K_I, are symmetric and positive-definite. Substituting (6) into (5) and using the

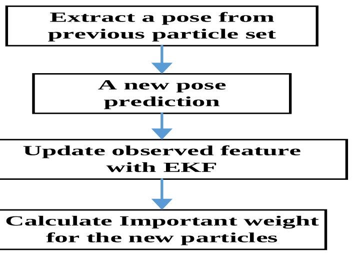

Extract a pose from previous particle set
A new pose prediction
Update observed feature with EKF
Calculate Important weight for the new particles

Fig. 4. Flowchart of the used FastSLAM 2.0 algorithm for each particle.

identity $\boldsymbol{J}\boldsymbol{J}^{+}=\boldsymbol{I}_3$ lead to the succeeding closed-loop error system

$$\begin{bmatrix}\dot{x}_e(t)\\ \dot{y}_e(t)\\ \dot{\theta}_e(t)\end{bmatrix}=\boldsymbol{J}(\theta(t))\boldsymbol{J}^{+}(\theta(t))\left(-\boldsymbol{K}_p\begin{bmatrix}x_e(t)\\ y_e(t)\\ \theta_e(t)\end{bmatrix}-\boldsymbol{K}_I\begin{bmatrix}\int_0^t x_e(\tau)d\tau\\ \int_0^t y_e(\tau)d\tau\\ \int_0^t \theta_e(\tau)d\tau\end{bmatrix}+\begin{bmatrix}\dot{x}_d(t)\\ \dot{y}_d(t)\\ \dot{\theta}_d(t)\end{bmatrix}\right)-\begin{bmatrix}\dot{x}_d\\ \dot{y}_d\\ \dot{\theta}_d\end{bmatrix} \quad (7)$$

$$=-\boldsymbol{K}_p\begin{bmatrix}x_e(t)\\ y_e(t)\\ \theta_e(t)\end{bmatrix}-\boldsymbol{K}_I\begin{bmatrix}\int_0^t x_e(\tau)d\tau\\ \int_0^t y_e(\tau)d\tau\\ \int_0^t \theta_e(\tau)d\tau\end{bmatrix}$$

where the globally asymptotical stability of the closed-loop error system can be easily proven by selecting the subsequent quadratic Lyapunov function

$$V_3(t)=\frac{1}{2}\begin{bmatrix}x_e(t) & y_e(t) & \theta_e(t)\end{bmatrix}\begin{bmatrix}x_e(t)\\ y_e(t)\\ \theta_e(t)\end{bmatrix}$$
$$+\frac{1}{2}\begin{bmatrix}\int_0^t x_e(\tau)d\tau & \int_0^t y_e(\tau)d\tau & \int_0^t \theta_e(\tau)d\tau\end{bmatrix}\mathbf{K}_\mathbf{I}\begin{bmatrix}\int_0^t x_e(\tau)d\tau\\ \int_0^t y_e(\tau)d\tau\\ \int_0^t \theta_e(\tau)d\tau\end{bmatrix} \quad (8)$$

2.3 **FASTSLAM 2.0**

In this subsection, FastSLAM 2.0 is adopted to address the SLAM problem of the MWOR. In the SLAM approach, the m^{th} particle $S_t^{[m]}$ contains the robot trajectory $s^{t,[m]}$ and N extended Kalman filter (EKF) to estimate N landmarks denoted by $\mu_{n,t}^{[m]}$ mean and covariance $\Sigma_{n,t}^{[m]}$ as below .

$$S_t^{[m]}=s^{t,[m]},\underbrace{\mu_{1,t}^{[m]},\Sigma_{1,t}^{[m]}}_{landmark\ \theta_1},\ldots,\underbrace{\mu_{N,t}^{[m]},\Sigma_{N,t}^{[m]}}_{landmark\ \theta_N}$$

Fig. 4 shows the flowchart of the used FastSLAM 2.0 algorithm for each particle. It is worth noting that the kinematic model of the MWOR is used for odometry model of the FastSLAM 2.0.

III. iBeacon Localization via Kalman Filtering

3.1 Kalman Filtering

Since the measured RSSI data from the iBeacon module are corrupted with white Gaussian noise processes, it is required to filter out the noise processes via a well-known Kalman filtering method. To this end, the used Kalman filter is with the following linear system model with the measurement equation.

$$\begin{aligned}x(k+1)&=x(k)+w(k)\\ z(k)&=x(k)+v(k)\end{aligned} \quad (9)$$

where $x(k)$ means the signal state; $w(k)$ the white Gaussian process noise and $w(k)\sim N(0,\mathrm{q(k)})$ where q(k) means the variance; $v(k)$ the white Gaussian measurement noise and $v(k)\sim N(0,\mathrm{r(k)})$ where $r(\mathrm{k})$ means the variance. Given the initial settings of $\hat{x}(0\,|\,0)$ and $p(0\,|\,0)$, the Kalman filtering algorithm is composed into the succeeding two steps:

Step 1: one-step time update or prediction whose two state and variance prediction equations are respectively given as blow.

$$\begin{aligned}\hat{x}(k\,|\,k-1)&=\hat{x}(k-1\,|\,k-1)\\ p(k\,|\,k-1)&=p(k-1\,|\,k-1)+q(k-1)\end{aligned} \quad (10)$$

Step 2: measurement or estimate update whose three updating equations are respectively described by

$$\begin{aligned}&K(k)=p(\mathrm{k}\,|\,\mathrm{k}-1)/(p(\mathrm{k}\,|\,\mathrm{k}-1)+r(\mathrm{k}))\\ &\hat{x}(\mathrm{k}\,|\,\mathrm{k})=\hat{x}(\mathrm{k}\,|\,\mathrm{k}-1)+K(\mathrm{k})\cdot(z(\mathrm{k})-\hat{x}(\mathrm{k}\,|\,\mathrm{k}-1))\\ &p(\mathrm{k}\,|\,\mathrm{k})=(1-K(\mathrm{k}))\cdot p(\mathrm{k|k\text{-}1})\end{aligned} \quad (11)$$

where $\hat{x}(k\,|\,k-1)$ denote the predicted state at time k-1; $\hat{x}(\mathrm{k}\,|\,\mathrm{k})$ is the estimated state at time k; $p(k\,|\,k-1)$ denote the predicted variance at time k-1; $p(\mathrm{k}\,|\,\mathrm{k})$ is the state estimate update at time k; $K(\mathrm{k})$ is the Kalman filtering gain at time k.

3.2 Triangulation Localization

This subsection is dedicated to investigate the triangulation localization of the MWOR using three iBeacon devices. Since the iBeacon receiver in the TX2 board receives three filtered RSSI (Received Signal Strength Indicator) signals from three different base stations, the received RSSI signals will encounter different levels of signal attenuation due to their different transmission distances. Through the degree of attenuation of the received signals, the distances between the receiver end and each individual base station can be calculated in the following equation.

$$d=10\,\hat{}\,((abs(RSSI)-A)/(10*\mathrm{n})) \quad (12)$$

where d is the calculated distance; $RSSI$ denotes the received signal strength; A represents the signal strength when the transmitter and receiver are separated by one meter; n is the environmental attenuation factor. The findings of both parameters, A and n, are referred to [12].

Next, determine the localization of the MWOR using the received RSSI signals. Assume that the given locations of the three iBeacon transmitters are located at (x_1,y_1,z_1) , (x_2,y_2,z_2) , (x_3,y_3,z_3) respectively. The goal of the triangulation localization is to find the unknown reference position coordinates by (x,y,z) using the three determined distances, d_1, d_2 ,and d_3 . As shown in Fig. 5, the three equations related to d_1, d_2, d_3 and (x,y,z) are given as

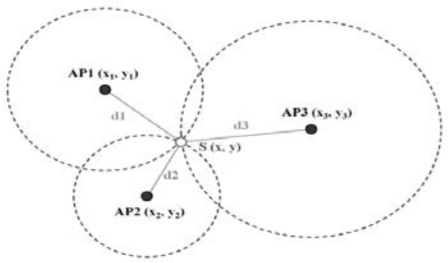


Fig. 5. Triangulation localization.

below.

$$\begin{aligned}(x_1-x)^2+(y_1-y)^2+(z_1-z)^2&=d_1^2\\(x_2-x)^2+(y_2-y)^2+(z_2-z)^2&=d_2^2\\(x_3-x)^2+(y_3-y)^2+(z_3-z)^2&=d_3^2\end{aligned}\qquad(13)$$

To solve for the current position, (x,y,z), of the MWOR, , one obtains the following simultaneous equation in the vector-matrix form

$$\mathbf{AX}=\mathbf{B}\qquad(14)$$

where $\mathbf{X}=[x\ y\ z]^T$;

$$\mathbf{A}=\begin{bmatrix}2(x_1-x_2) & 2(y_1-y_2) & 2(z_1-z_2)\\ 2(x_2-x_3) & 2(y_2-y_3) & 2(z_2-z_3)\\ 2(x_3-x_1) & 2(y_3-y_1) & 2(z_3-z_1)\end{bmatrix}$$

$$\mathbf{B}=\begin{bmatrix}d_2^2\text{-}d_1^2+(x_1^2-x_2^2)+(y_1^2-y_2^2)+(z_1^2-z_2^2)\\ d_3^2\text{-}d_2^2+(x_2^2-x_3^2)+(y_2^2-y_3^2)+(z_2^2-z_3^2)\\ d_1^2\text{-}d_3^2+(x_3^2-x_1^2)+(y_3^2-y_1^2)+(z_3^2-z_1^2)\end{bmatrix}$$

From (14), the current global position, (x,y,z), of the MWOR is found by

$$\mathbf{X}=\mathbf{A}^{-1}\mathbf{B}\qquad(15)$$

V. Self-Driving To Avoid Obstacles Uisng SegNet

4.1 Introduction to SegNet

SegNet has an encoder network and also has a corresponding decoder network, followed by a final pixelwise classification layer. The architecture is illustrated in Fig. 6 The encoder network makes up of 13 convolutional layers which correspond to the first 13 convolutional layers in the VGG16 [5] network designed for object classification.Therefore, we can initialize the training process fromweights trained for classification on large datasets. Also we can discard the entirly connected layers in favour of retaining higher resolution feature maps at the deepest encoder output. This also reduces the number of parameters in the SegNet encoder network significantly as compared to other recent architectures [6]. Every encoder layer has a corresponding decoder layer. Hence, the decoder network has 13 layers. The last decoder output is fed to a multi-class soft-max classifier in order .to produce class probabilities for every pixel independently.

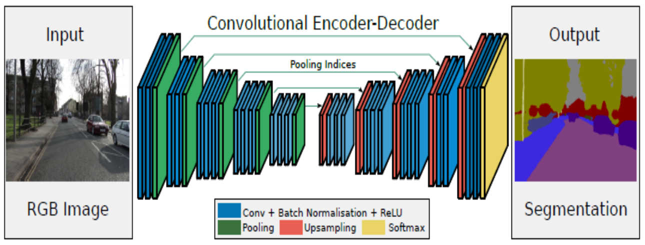


Fig. 6. Architecture of SegNet.

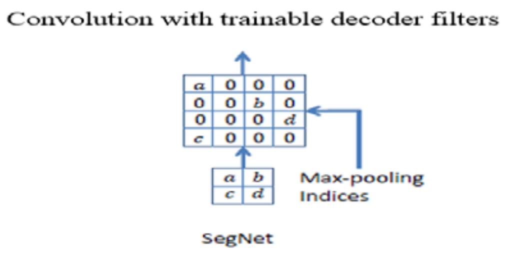


Fig. 7. Decoders of the SegNet.

Every encoder in each encoder network shows convolution with a filter bank to produce a set of feature maps. These are then batch normalized [7],[8].Then an element-wise rectifiedlinear non-linearity (ReLU) max is applied. Following that, max-pooling with a 2 × 2 window and stride 2 is performed and the resulting output is sub-sampled by a factor of 2. Max-pooling is used to achieve translation invariance over small spatial shifts in the input image. Sub-sampling results in a large input image context for each pixel in the feature map. While several layers of max-pooling and sub-sampling can achieve more translation invariance for robust classification correspondingly. There is a loss of spatial resolution of the feature maps. This increasingly lossy image representation is not beneficial for segmentation where boundary delineation is vital. Hence, it is necessary to catch and store boundary information in the encoder feature maps before sub-sampling is expressed. If memory during inference is not constrained, all the encoder feature maps can be stored well-done. The map is sometime not the case in practical applications. Hence, we propose a efficient way to store this information. It involves storing only the max-pooling indices,i.e, the locations of the maximum feature value in each pooling window is memorized for each encoder feature map. In principle,the map can be done using 2 bits for each 2 ×2 pooling window.Therefore, it is more efficient to store as compared to memorizing feature maps in float precision. This can lower memory storage results in a slight loss of accuracy.However, it is still suitable for practical applications.

This appropriate decoder in the decoder network upsamples its input feature maps using the memorized max-pooling indices from the corresponding encoder feature maps. The step produces sparse feature maps [9]. The SegNet decoding technique is illustrated in Fig. 7. These feature maps are then convolved with a trainable decoder filter bank to produce dense feature maps.A batch normalization step is then applied to each of these maps.Note that the decoder

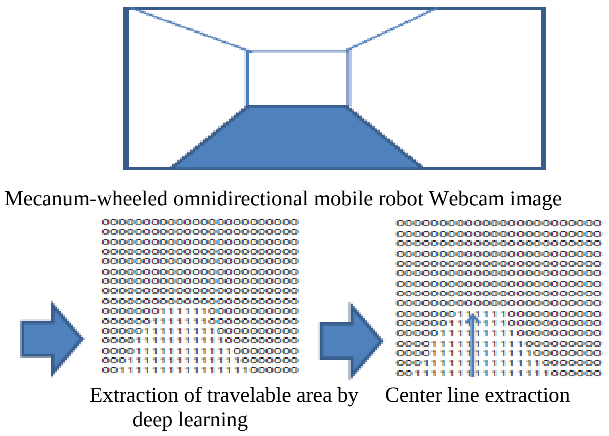

Mecanum-wheeled omnidirectional mobile robot Webcam image
Extraction of travelable area by deep learning
Center line extraction

Fig. 8. Flowchart of the image recognition.

corresponding to the first encoder produces a multi-channel feature map, although its encoder input has 3 channels (RGB). It is not like the other decoders in the network which produce feature maps with the same number of size and channels as their encoder inputs.The huge dimensional feature representation at the output of the final decoder is fed to a trainable soft-max classifier. The soft-max classifies every pixel independently. The output of the soft-max classifier is a K channel image of probabilities where K is the number of classes. The predicted segmentation corresponds to the class with maximum probability at every pixel [10].

4.2 Estimation of Travelable Area

This subsection will briefly describe the system architecture and methods of the autonomous navigation to avoid any static and dynamic obstacles. The self-driving procedure to prevent any collisions from obstacles is described in the following six steps. First, the image recognition is done by using the flowchart in Fig.8. As depicted in Fig. 9, The webcam images are processed and extracted by SegNet via the Jetson TX2 board, in order to find a travelable area. Second, calculate the determined centerline; Fig.10 shows how to find the centerline (locations represented as a "1" in Fig. 10). The centerline is extracted from the travelable area using the least squares method by finding the centers of the widths of the travelable area; afterwards, a straight line is then obtained along these centers with the least error, as shown in Fig. 11(c). Third, calculate the orientation deviation between the centerline and the current orientation of the mobile robot. Forth, generate the motion commands. After the positions of the tip of the centerline and the front of the Mecanum-wheeled omnidirectional mobile robot are mapped to two-dimensional space, the deviation between the centerline and the direction in which the mobile robot is facing is calculated by the Jetson TX2 board, and send the feedback commands to mobile robot so as to reduce the deviation from the centerline. Fifth, calculates the current position of the robot via the iBeacon location system and LiDAR. Sixth, generates motion control commands to activate the four servomotors for autonomous navigation.

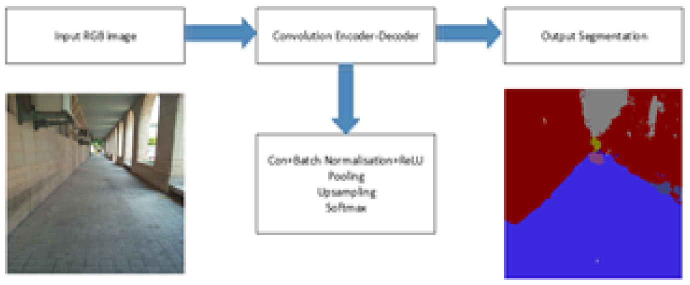
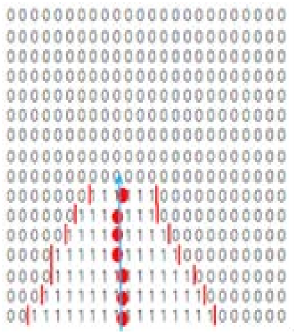


Fig. 9. Image segmentation process of SegNet.



(1) Find the center point in the width direction in the travelable area
(2) Find a straight line with the least error from the center point. (Least squares method)

Fig. 10. Centerline calculation

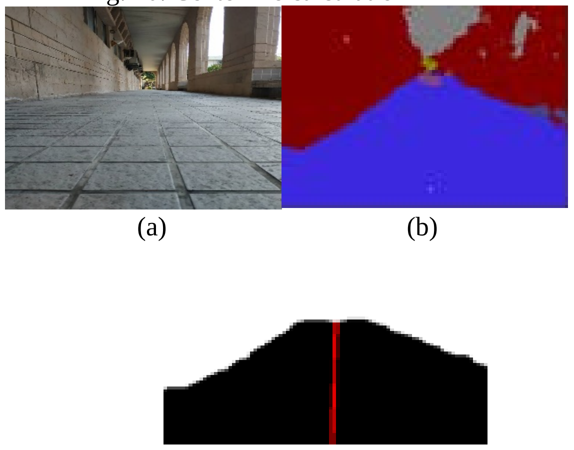


Fig. 11. Segmentation results using SegNet and centerline detection: (a) an original picture; (b)segmentation results using Segnet; (c) detected centerline of travelling area.

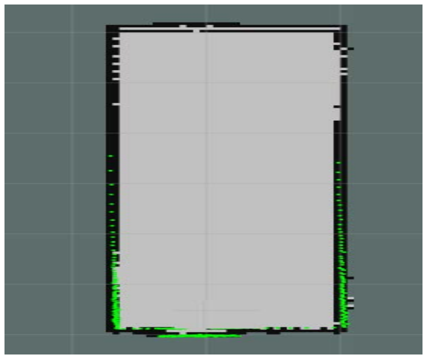


Fig. 12. Map created by Using the FastSLAM 2.0 algorithm.

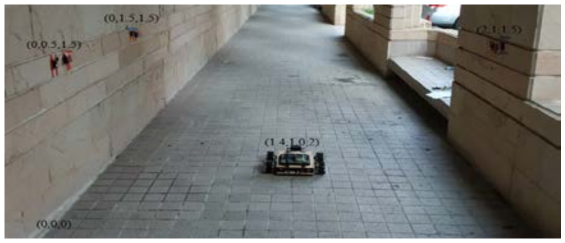


Fig. 13. Triangulation Localization.

TABLE II. Average measurement errors of the used iBeacon module in the x and y frames.

item / position	Average measurement obtained from the iBeacon module	True value	Average Error
X (m)	1.8(m)	1.45(m)	0.35(m)
Y (m)	2.9(m)	1.95(m)	0.95(m)

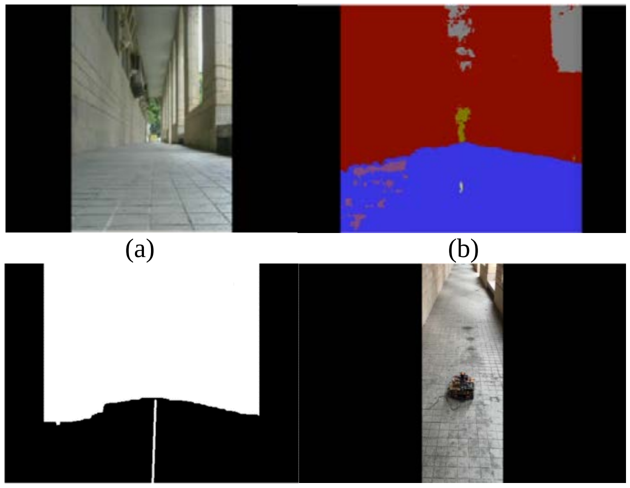


Fig.14. Experimental results: (a) Initial experimental environment; (b) segmentation of SegNet;(c) centerline of traveling area;(d) robot experiment results.

VI. Experimental Results and Discussion

This section will conduct three experimental results to show the effectiveness of the proposed method. The first experiment is aimed to test the experimental MWOR under ROS. Fig. 12 shows the environmental map created by using the FastSLAM 2.0 algorithm, showing the success of the system integration for the experimental MWOR.

The second experiment is carried out to show the localization performance of the used iBeacon devices. Fig. 13 displays the installation of the three iBeacon devices where the receiver is the Jetson TX2 board. The computed location and average errors of the WMOR were respectively shown in Fig. 13 and Table 2, showing that the iBeacon localization is helpful in localizing the MWOR during its autonomous navigation.

The third experiment is performed to examine the effectiveness of the self-driving method using the experimental set-up in the subsequent steps. At the outset, Fig. 14(a) shows the original image captured by the on-board webcam. Second, the captured image was transferred to the TX2 computing module, in order to achieve the image segmentation via SegNet, as shown in Fig.14(b). Third, the simple binary method and least squares method were employed to find the centerline of the travelable area as Fig. 14(c) shows. At last, the proposed trajectory tracking was exploited to control the Mecanum-wheeled omnidirectional mobile robot to move along with the desired centerline as shown in Fig. 14(d), in order to move toward its destination.

VII. CONCLUSIONS AND FUTURE WORK

This paper has presented an autonomous navigation system for a Mecanum-wheeled omnidirectional mobile robot (MWOR) based on image recognition by deep learning and the designed localization module via the iBeacon signals and LiDAR. Preliminary experimental results confirmed that this system could autonomously run the MWOR to move along the centerline of its current environment, thereby avoiding static and dynamic obstacles. In the future, we aim to combine SegNet and an optimal path planning method to carry out a more complete autonomous navigation in an indoor environment.

Acknowledgements

The authors deeply acknowledge financial support from the MOST, Taiwan, ROC, under contract MOST 107-2221-E-005 -073-MY2.

REFERENCES

[1] C. C. Yu, C. F. Hsu, and F. C. Tai, “A laboratory course on mobile robotics education, “iRobotics, vol. 1, no.4, pp.37-43, December 2018..

[2] J. Tang, J. Li, and X. Xu, "Segnet-based gland segmentation from colon cancer histology images," 2018 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC), Nanjing, 2018, pp. 1078-1082.

[3] S. Quinlan and O. Khatib, "Elastic bands: connecting path planning and control," *[1993] Proceedings IEEE International Conference on Robotics and Automation*, Atlanta, GA, USA, 1993, pp. 802-807 vol.2. doi: 10.1109/ROBOT.1993.291936.

[4] G. Ros, L. Sellart, J. Materzynska, D. Vazquez and A. M. Lopez, "The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes," *2016 Int. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016, pp. 3234-3243.

[5] S. Isobe and S. Arai, "Inference with model uncertainty on indoor scene for semantic segmentation," in *Proc. 2017 IEEE Global Conf. on Signal and Information Processing (GlobalSIP)*, Montreal, QC, 2017, pp. 1170-1174.

[6] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[7] J. Long ,E.Shelhamer,and T. Darrell,”Fully convolutional networks for semantic segmention,”in CVPR,pp.343331-3440,2015.

[8] S. loffe and C. Szegedy,”Batch normailization :Accelerating deep network training by reducing internal covariate shift,” CoRR,vol. abs/1502.03167,2015.

[9] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481-2495, Dec. 1 2017.

[10] D. Naik and C. D. Jaidhar, "Image segmentation using encoder-decoder architecture and region consistency activation," *2016 11th International Conference on Industrial and Information Systems (ICIIS)*, Roorkee, 2016, pp. 724-729.

[11] S. Isobe and S. Arai, "Inference with model uncertainty on indoor scene for semantic segmentation," *2017 IEEE Global Conf. on Signal and Inform. Processing (GlobalSIP)*, Montreal, QC, 2017, pp. 1170-1174.

[12] L. Reger, "Securely connected vehicles - what it takes to make self-driving cars a reality," *2016 21th IEEE European Test Symposium (ETS)*, Amsterdam, 2016, pp. 1-1.