

Design and Implementation of Augmented Reality Unmanned Vehicle

Chia-Hsin Cheng*, *Member IEEE*, Tao-Ping Wang, Feng-Cheng Wu, and Hou-Ming Chen
Department of Electrical Engineering National Formosa University Yunlin County Taiwan

Abstract—This paper uses Arduino Yun on YBB Carriers to set up the server, and then transmits the picture taken by the YBB Car camera to the computer via Wi-Fi. In order to get the location information, we use the computer to progress image processing. After computing, the computer will send command to control the Automatic guide vehicle (AGV). At the same time, the related information will display at the user side. It will help users to get the information of AGV's location even they are not at the presence.

I. INTRODUCTION

Due to the improvement of sensing technology, application of Internet of Things (IoT) technology and information technology, Automatic guide vehicle (AGV) has also developed into a smart van. In recent years, in the fully automated smart logistics center, unmanned mobile vehicles as an important part of the IoT, to become an intelligent logistics terminal, make the administrator easier to manage the warehouse [1]. Through the technology of image recognition and algorithm, AGV can improve the autonomous judgment ability and achieve the goal of “fully automated”. For example, truck drivers simply enter the unloading and loading information on their mobile phones after arriving in the warehouse, and the AGV will use the images to identify the information on the trucks and warehouses. The AGV will load and unload cargos automatically, at the same time it will upload information of cargos to the cargo control system. It can decrease the risk of human errors, and all the information can upload instantly, it can also record the status of the cargos through the camera on the vehicle to facilitate the management of managers in the process. When in transit the AGV will through the image to determine the transport vehicle offset, Hinder or the cargo drops. Therefore, this paper will integrate technologies related to Arduino, servomotors, image recognition, algorithms and so on [2-3] to control the vehicle remotely to a designated location by using a wireless network. If the vehicle is offset, Obstruct or cargo drop, an exceptional condition signal will transmitted through the wireless network, user can through the computer terminal to observe the status quo, so that back-end administrators can be rush to the scene as soon as possible.

II. RELATED WORK

A. System Architecture

In the experiment, we employ Wi-Fi to connect the computer and the AGV. AGV will send image taken from the camera. After computer side receive images and execute back-

end image processing, it can identify lane lines and street signs which are place at corners to determine whether to reach the corner or deviate from the lane [4], and plan a timely path or notify staff needs Exclusion of roadblocks. The overall system architecture shown in Figure 1.

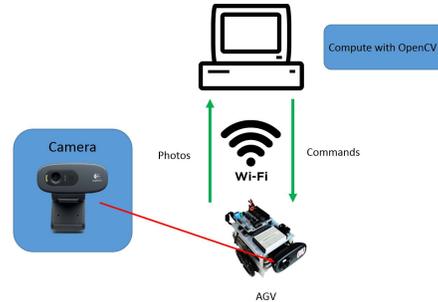


Fig. 1. System architecture diagram

B. Path Planning Principles

In this paper, the experimental map set up each intersection a set of fixed coordinates. Using the way of hierarchy to find the best path from the starting point to the terminal point, the path will present as a set of coordinates array. As long as the AGV move to next coordinates, it will reach destination at the end. The experimental map shown in Figure 2.

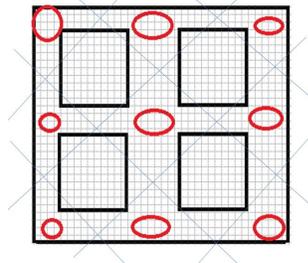


Fig. 2. Experimental map

III. IMAGE PROCESSING

A. Image Processing Architecture

This paper uses the image captured from camera on AGV, the computer-side use OpenCV with Visual Studio to identify lane lines and graphics. After the image is gray-scaled, the Gaussian filter is use to eliminate the noise, and then the Canny image edge detection algorithm is used to determine whether there is deviation from the lane. The corner illustration uses the feature point detection to identify the shape of the figure.

B. Image Filter

In this paper, we use Gaussian filter, which is a filter whose impulse response is a Gaussian function. Gaussian filters have

the properties of having no overshoot to a step function input while minimizing the rise and fall time.

Gaussian function is define as two-dimensional space:

$$G(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{(u^2+v^2)}{2\sigma^2}} \quad (1)$$

Which σ is standard deviation of normal distribution.

C. Edge Detection

Canny edge detection is a multi-level edge detection algorithm. First, filters the noise in the image by Gaussian filter. The main reason for choosing a Gaussian filter is that the processing result is better than the average filter.

Second, we look for the brightness gradient in the image. Because the higher brightness gradient could be the edge of the image. However, each image has different brightness and environment, so we need to use two thresholds to distinguish.

Finally, a larger threshold is use to mark the edges of the image before tracking, and another smaller threshold is used to track the more obscure part of the edge [5].

D. Comparison of feature points

Find similar items in two images. First, you need to do the interest point detection which relevant points in both image, and these points are usually unique, such as the corner, T-shaped intersection, etc. Second, collect information about the point around the interest point and form a feature vector, this vector is call a descriptor. Finally, compare the descriptors, and the distance between descriptors can be use as the basis for comparison.

Feature point detection is use to identify the image, and feature point matching uses Speeded-Up Robust Features (SURF). We use the SURF algorithm is because it has less disturbed by the size, plane rotation, and so on. The processing speed is also faster than Scale-Invariant Feature Transform (SIFT) algorithms [6-7].

IV. EXPERIMENT RESULTS

First, set the Wi-Fi connection and burning pre-programmed Arduino Yun PCB board then put the AGV to the initial position. Second, open the computer-side OpenCV with Visual Studio, connected to the AGV by Wi-Fi and receive the image returned by the camera. Third, enter destination then Visual Studio plans the path and begins to walk. The AGV will capture the signs (roadside square signpost) then the image will compare with the database and gives the next instruction. The driving process will continue to make the path correction, as shown in Figure 3, until the AGV arrives at the destination. Figure 4 shows AGV walking conditions and augmented reality images on Visual Studio [8].

V. CONCLUSIONS

This paper for AGV follow the lane to move forward, identify the sign, user can know the location and view the scene by shoot in front. Although there are still some problems to be solved in the future. If concepts and technologies of this

system can be applied in factories. Users can monitor at the back-end (not necessarily go to the site) which will reduce much labor costs. Due to the issue of delay caused by the transmission of images via Wi-Fi, the system does not have the ability to respond immediately. Expect in the future, we can compress the processing time under 0.2 seconds to achieve real-time reaction. To enhance the processing speed by add other sensor-assisted, detect obstacles on the road to automatically avoid obstacles and achieve more AGV control.

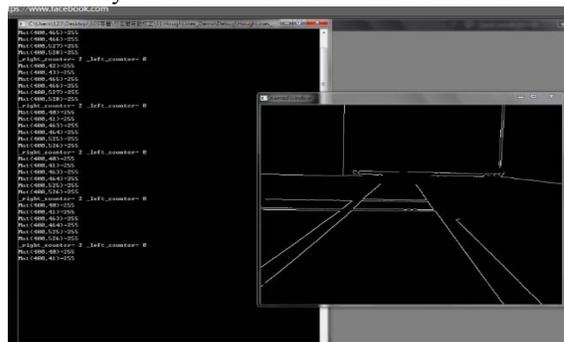


Figure 3. Route planning and lane correction

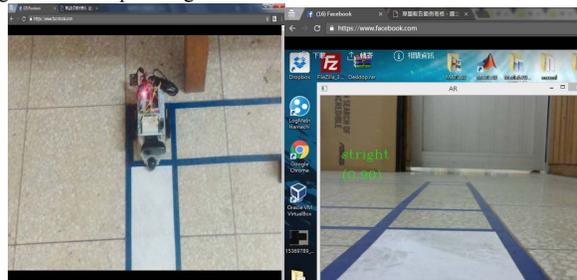


Figure 4. The actual walking conditions and current position with augmented reality

REFERENCEFS

- [1] Koushik Sinha, Sasthi C. Ghosh, Bhabani P. Sinha, *Wireless Networks and Mobile Computing*, Chapman and Hall/CRC 2015.
- [2] Rafael C. Gonzalez, Richard E. Woods, *Digital image processing*, 3th Edition, 2006.
- [3] Harald Schmidt, *OpenCV Basiswissen*, 2013.
- [4] P. Shopa, N. Sumitha and P. S. K. Patra, "Traffic sign detection and recognition using OpenCV," International Conference on Information Communication and Embedded Systems (ICICES2014), Chennai, 2014, pp. 1-6.
- [5] J. Wang and J. Watada, "Panoramic image mosaic based on SURF algorithm using OpenCV," 2015 IEEE 9th International Symposium on Intelligent Signal Processing (WISP) Proceedings, Siena, 2015, pp. 1-6.
- [6] F. K. Noble, "Comparison of OpenCV's feature detectors and feature matchers," 2016 23rd International Conference on Mechatronics and Machine Vision in Practice (M2VIP), Nanjing, 2016, pp. 1-6.
- [7] F. Guan, X. Liu, W. Feng and H. Mo, "Multi target recognition based on SURF algorithm," 2013 6th International Congress on Image and Signal Processing (CISP), Hangzhou, 2013, pp. 448-453.
- [8] R. S. Deepthi and S. Sankaraiiah, "Implementation of mobile platform using Qt and OpenCV for image processing applications," 2011 IEEE Conference on Open Systems, Langkawi, 2011, pp. 284-289.