An End-to-end Auto-driving Method Based on 3D Lidar

H Dong[1], M Wang[1], W Zhang[1], W Shu[1], C Chen[1], Y Z Lu[2] and H F Li[2]

[1]Tsinghua Suzhou Automotive Research Institute (Wujiang), Suzhou, China.
[2]SAIC Motor Corporation Limited, Shanghai, China.

Keywords: Lidar, Deep Learning, Auto-driving method

Abstract. The development of artificial intelligence, especially deep learning engineering technology, has made auto-driving cars more and more realistic. The end-to-end auto-driving method is an automatic driving system which is different from the rule-based system of. It uses the data from the environment to output vehicle control information solutions directly, greatly reducing the system complexity. 3D Lidar is the core sensor of automatic driving system. In this paper, a deep convolution neural network is designed for the end-to-end automatic driving method. This paper uses 64 -line 3D Lidar data and transformation algorithm, transforms the 3D Lidar point-cloud data into depth images which can be used directly by an end-to-end deep learning network. This paper matches 3D Lidar data with vehicle-mounted Can bus data to obtain data and tags which will be feed into the deep learning network. The output of the deep learning network is the controlling information that directly acts on the vehicle. Based on experimental verification, the end - to - end automatic driving method based on 3D Lidar is of great value and potential for further development.

1. Introduction

With the development of economy, the increase of vehicle ownership and the number of non-professional drivers, the occurrence of traffic accidents is more frequent; traffic accidents have become a major public hazard in modern society. 90% of traffic accidents related to drivers are caused by drivers' inattention. In this case, the development of intelligent vehicles with autonomous driving functions by using high and new technologies has become one of the key means to solve the problem and a key component of intelligent transportation.

According to different technical routes, the automatic driving system includes rule-base automatic driving system and end-to-end automatic driving system. In a rule-based automatic driving system, the entire vehicle is closed loop system. The information spreads in vehicle-sensors-perception-model - decision-vehicle. Vehicles understands the surrounding the whole scene before making decisions; This system involves a lot of related problems, needs for artificial disassemble, such as traffic detection, lane line detection, traffic signal recognition, and regional identification, etc. The rule-based automatic driving system is very large, and the rule construction is complex, which leads to a huge amount of on-board computer computation and a high demand for hardware.

In an end-to-end autopilot system, the environmental data is input automated driving core processing system (mainly refers to the deep learning system) as an end, and deep learning system outputs signals to control vehicle directly as the other side; the system does not need to manually dismantling tasks, driving behavior is very similar to human driving behavior; The basic idea is to establish and simulate the neural network[1][2] of human brain for analysis and learning, and to interpret data by training the network and imitating the mechanism of human brain. Compared with the traditional rule-based method, this method has great research and engineering potential in solving the high dynamic, uncertain, multi-scene, strong nonlinear problems.

Both rule-based automatic driving system and end-to-end automatic driving system need sensors to perceive the environmental data of the vehicle and process the environmental data through on-board computer. At present, vehicle-mounted environmental perception sensors mainly include monocular camera, binocular camera, 3D Lidar and millimeter wave radar, etc. Among these sensors, 3D Lidar has a large data volume, a high ranging accuracy, 3D environment modeling and all-weather work can be carried out, so it is the most ideal sensor for automatic driving.

This paper transforms 64-line 3D Lidar data to a depth image, and matches the vehicle Can bus signal which collected at the same time. The depth image of the 3D Lidar graph is treated as the input data, and the Can bus signals which includes the steering wheel angle and speed as labels. They are input to the vehicle deep learning system. The system outputs vehicle control signals, directly controls the movement of the vehicle, and completes the construction of the whole end-to-end automatic driving system.

2. Environment Data Processing in Automatic Driving System

2.1. The Transformation Algorithm From 64-line 3D Lidar Point-cloud Data to Depth image

Figure 1 shows Velodyne HDL-64E Lidar, which is a civilian 3D Lidar the highest precision on the market at present.

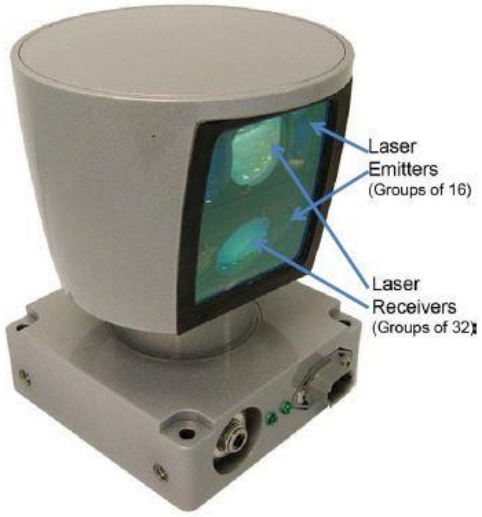


Figure 1. Velodyne HDL-64E 3D Lidar.

The data points from Velodyne HDL-64e are very dense, with more than 1,800 data packets per second. Therefore, the data obtained is called Point cloud data. Point cloud is a collection of data points in three-dimensional space. Each data contains the discrete coordinates of the actual target relative to the 3D Lidar light source and the light intensity information of the point. Through the processing of point cloud data, the basic geographic three-dimensional information of the surrounding environment can be obtained quickly and accurately.

Depth image, also known as range image, refers to the image with pixel value as the distance (depth) from the image collector (3D Lidar, camera, etc.) to each point in the scene[3]. Depth image directly reflects the geometry of the visible surface in the environment and is a good choice for the input image of end-to-end convolutional neural network. Depth image can be calculated as point cloud data after coordinate transformation, and point cloud data with rules and necessary information can also be converted into depth image. In this paper, 3D Lidar point cloud data processing is to design the algorithm of turning depth image from the point cloud data acquired by 3D Lidar[4], and to generate the depth image of the driving environment to serve as the input of end-to-end neural network. The algorithm flow of generating a depth image from a frame of 3D Lidar point cloud data is shown in figure 2.

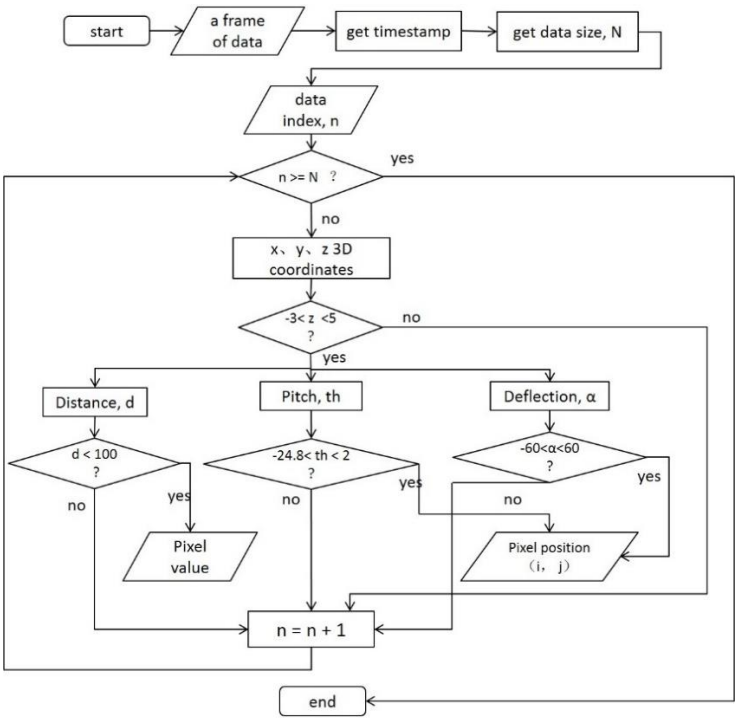
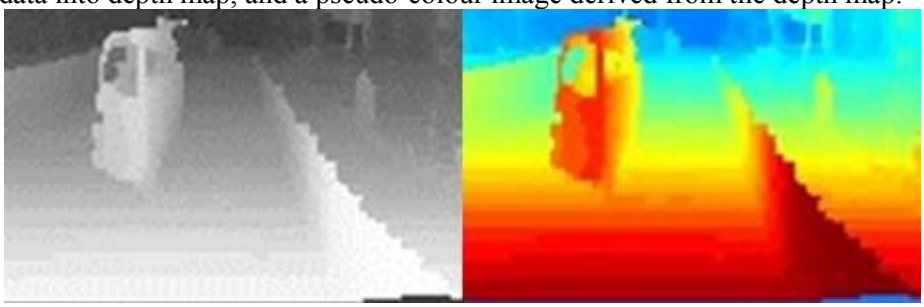


Figure 2. the algorithm flow of generating a depth image from a frame of 3D Lidar point cloud data.

The data structure of a complete 3D Lidar data frame can be divided into three parts: the timestamp of the data in the frame, the amount of data in a frame (that is, the number of points containing complete 3d coordinates, which may vary in different data frames), and the coordinates of each point. Therefore, in the algorithm flow, the timestamp and data volume should be resolved first. The x, y and z coordinates of each point are obtained successively[5]. According to the three-dimensional coordinates, the distance from the point to the image collector (the fixed point of the 3D Lidar light source), the elevation angle of the point relative to the horizontal plane, and the deflection angle of the point relative to the longitudinal section plane are calculated. These data serve as a criterion for whether the point becomes a valid point of depth map. According to the discriminant conditions of each criterion and combined with the projection algorithm, the pixel position and pixel value corresponding to this point on the depth map are obtained[6].

Figure 3 shows a depth map obtained according to the above algorithm of converting 3D Lidar point cloud data into depth map, and a pseudo-colour image derived from the depth map.



(a) (b)

Figure 3. a depth map (a) and its pseudo-color image (b).

2.2. Precise Synchronous Matching Between Different Data

In the end-to-end autonomous driving method based on 3D Lidar, the deep learning system that outputs automobile motion information is a supervised learning system. For each depth image generated from 3D Lidar point cloud data, a "label" is required to correspond to it. Therefore, a key

step in this paper is to ensure the precise synchronization and matching between 3D Lidar data, driver operation data and other types of data output by the car Can bus, that is, to ensure that each pair of output and input of the neural network are at the same time point. In order to ensure accurate matching, any data frame in the various types of data obtained has a timestamp, which is locally timed by the data acquisition computer (PC) to the millisecond. The synchronization of timestamps can be used to synchronously match the data.

The original timestamp format for all types of data is Beijing time, in the format of "year-month-day hour: minute: second. Millisecond". Each formatted time should be timestamped to facilitate the comparison of time stamp sequence before matching. Among all kinds of data, 3D Lidar has the lowest acquisition frequency and the smallest amount of data. Therefore, the time stamp of 3D Lidar data is taken as the benchmark for matching. The flow chart of matching algorithm is shown in figure 4.

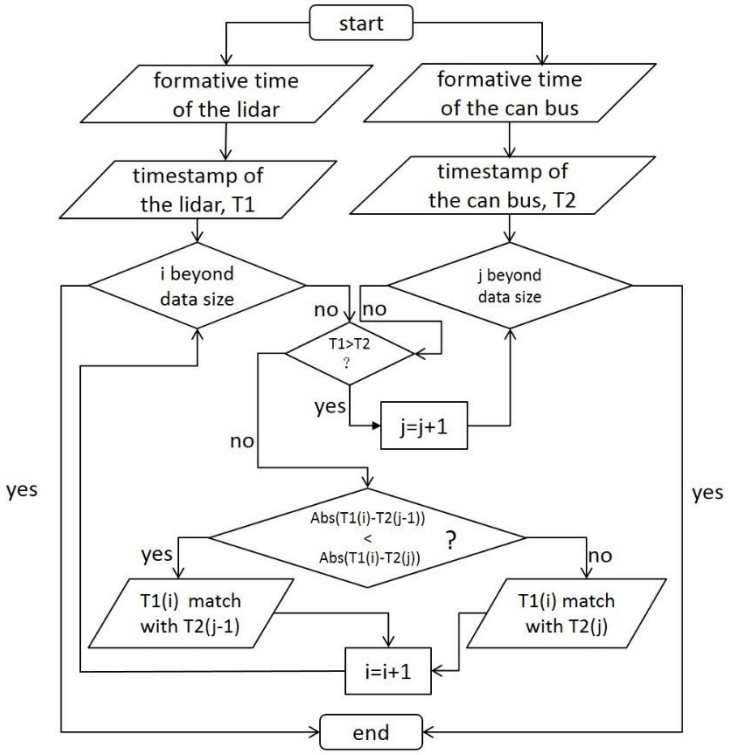


Figure 4. the flow chart of data synchronization algorithm.

In this algorithm, two timestamps on both sides of the 3D Lidar timestamp are first found in the timestamp data of the Can bus; then the two timestamps are compared to see which one is closer to the above motioned 3D Lidar timestamp. The data of the frame in which the timestamp is closer is the synchronization data matching with the 3D Lidar data frame.

3. The End-to-end Deep Learning Architecture Design and Model Training

3.1. Architecture Design of Deep Learning System in End-to-end Autonomous Driving Method

In the 3D Lidar-based end-to-end auto-driving method, the deep learning training system which is based on the depth convolutional neural network (the DCNN) is a key role. The system receives 3D Lidar depth images and driver's controlling data from the Can bus as training data and outputs vehicle controlling information. The framework of the training system is shown in figure 5.

The data collected by the 3D Lidar are transformed to obtain the depth image that can be used as the input of the deep convolutional neural network and fed into the neural network with a designed structure. The output of the deep convolutional neural network is a characterization of the steering wheel angle, which is compared with the actual steering wheel angle that is the label of the input image[7]. The obtained difference values are propagated back to the neural network with the Back-

Propagation algorithm (BP) to adjust the parameters of the deep convolutional neural network. The weight of the network parameter is adjusted to make its actual output closer to the desired output, that is, the steering wheel angle output as the label[8].

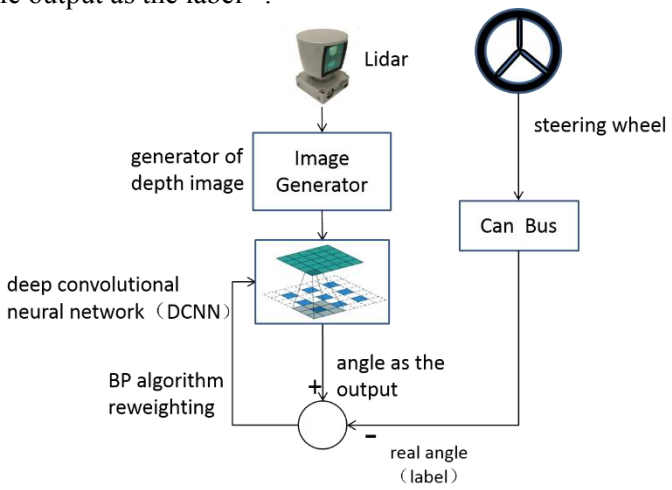


Figure 5. the block diagram of the end-to-end automatic driving training system.

3.2. Acquisition and Enhancement of The Training Data

The training data in this paper were collected from the "067" electric vehicle developed by Suzhou automotive research institute of Tsinghua university under various road and weather conditions. This vehicle whose vehicle-mounted wire control equipment was designed and modified is equipped with a Velodyne HDL-64e 3D Lidar supplied by SAIC Motor. The roads in which the data was collected are mainly in the park of Suzhou automotive research institute of Tsinghua university. The environment elements include single-lane or two-lane roads which are marked with lane lines, and residential roads with parked vehicles. In the process of data collection, the vehicle travels at a speed of 30~50 Km/h, and the driver drives the vehicle in a normal state. After the collection of multi-section, multi-scene and multi-weather condition, a total of about 10 hours of training data were obtained. The test vehicle equipped with the 64-line 3D Lidar is shown in figure 6.

Figure 6. the test vehicle equipped with the 64-line 3D Lidar.

In the training of an end-to-end deep neural network for automatic driving, insufficient data volume, complex driving environment and other factors may lead to insufficient feature extraction and overfitting of the neural network, so the data should be enhanced and expanded. The amount of the data is increased by adding artificial bias, rotation, flip, mirroring and other image processing schemes, so the network can be taught how to extract effective features from an adverse position or direction. These perturbation-like operations are randomly selected from a normal distribution whose mean and

standard deviation are also designed to ensure that the resulting picture is still within the normal, human-understandable range.

3.3. Structural Design of Deep Convolutional Neural Network

Based on the open source deep learning development framework tensorflow-keras, this paper designed a deep convolutional neural network structure suitable for end-to-end autonomous driving tasks based on 3D Lidar, as shown in figure 7.

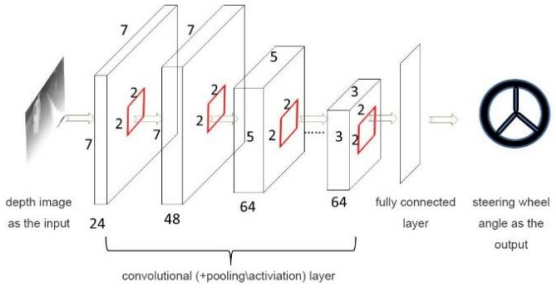


Figure 7. the structure of the end-to-end deep convolutional neural network.

In the deep-learning convolution neural network, the convolution layer is used to perform the most important core processing tasks - feature extraction. An important feature of deep-learning convolutional neural network is that it has many layers, many parameters and various configurations. The parameter configuration (number of convolution layers, size of convolution kernel, etc.) of the convolution layer in this paper is determined through a series of configuration experiments. The convolutional neural network designed in this paper USES three kinds of convolution kernel, the 7×7 kernel, the 5×5 kernel and the 3×3 kernel. For each convolution layer, in addition to the convolution kernel, the number of convolution layers and the step size of the convolution operation on the picture are determined. In fact, in the convolutional neural network shown in figure 7, each convolutional layer is "generalized", including not only the convolutional layer, but also the pooled layer (down-sampling layer) and the activation layer (also known as the nonlinear layer) following it. The 2×2 square in each convolutional layer in the figure represents the down-sampling layer. In the convolutional neural network designed in this paper, the pooling method adopts max pooling, and the Relu function is used for activation.

The convolution layer is followed by several (only 1 is shown in FIG. 7) fully connected layers (fc). The last fully connected layer outputs a scalar value, namely the steering wheel Angle value. In this article, the full connection layer is actually designed as a steering controller. In the actual vehicle control process, the output of the fully connection layer is directly input to the vehicle-mounted wire control unit to control the lateral movement of the vehicle. In fact, in the network structure of this paper, the part as the feature extractor (convolutional layers) and the part as the controller (full connection layers) are fused together and are not completely independent and separable.

3.4. Training and Evaluation of Deep Convolutional Neural Networks

After the convolutional neural network structure is established, the training data and its labels are fed to the network to start the network training. The core network training processor in this article is NVIDIA Graphics Processing Unit (GPU), GeForce GTX 1070.

Loss function is the most important indicator to evaluate the training effect of the deep convolution neural network, and it is used to estimate the degree of inconsistency between the predicted values and real values which actually are the labels corresponding to the input images[9], and it is a nonnegative real function. The closer this value is to 0, the better the training effect of the model and the better the robustness of the model. According to the design in this paper, both the model output and label are one-dimensional scalars, so the mean-squared error (MSE) between them is selected as the loss function of the model. The loss function formula is as follows:

$$L(\tilde{y}, y) = \sum_i (\tilde{y}_i - y)^2 \quad (1)$$

In the formula, ‘L’ is the loss function and ‘i’ represents the number of the picture in the training process.

The training of deep-learning model needs to select specific optimization methods[10]. Different optimization methods may lead to different convergence rates of loss functions. Adaptive moment estimation (Adam) algorithm is selected in this project. Its advantages are that it has a small demand for hardware memory. Different adaptive learning rates are calculated for different parameters. After the bias correction, the learning rate of each iteration has a certain range to make the parameters stable.

In the actual training process, batch processing should be carried out for the input data. The number of input images each time is called batch_size. At the same time, the label value should be normalized. According to the steering wheel angle, the normalized coefficient is 500.

After the above network structure design and relevant training parameters configuration, the model is trained with the collected data. After several iterations, the training time reaches 20 hours, and the loss function of the training process model is shown in figure 8.

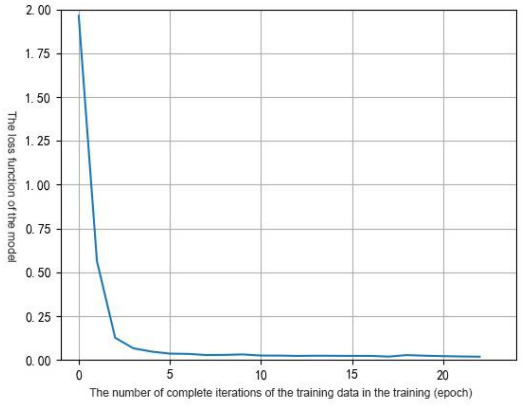


Figure 8. the curve of the model loss function during training.

Figure 8 shows the number of complete iterations of the training data in the training process on the horizontal axis and the loss function of the model on the vertical axis. According to the results shown in the figure, at the beginning of the training, the loss function was large. After the back propagation and the optimization method, the loss function rapidly decreased and finally stabilized at 0.017. According to batch_size and the normalized coefficient of label value, the mean square error between the steering wheel angle value output by the deep learning model and the actual value was calculated as:

$$0.017*500/64 = 0.133^\circ \quad (2)$$

This value has reached the real-time control precision of the driver on the steering wheel during driving.

3.5. Automobile Test Result

After the training of the end-to-end convolutional neural network model was completed, the test vehicle with Velodyne HDL-64e 3D Lidar as shown in figure 6 was used for the end-to-end automatic driving test. Both the deep learning model and the 3D Lidar acquisition software work on Windows computers. The acquisition frequency of the 3D Lidar is 10 frames per second, and the running speed of the deep learning model on the working PC is also up to 10 frames per second.

The vehicle test was conducted in the intelligent network vehicle test demonstration area of Suzhou automotive research institute of Tsinghua university. The vehicle travelled at a speed of 25 km/h, with a total mileage of 2.5 km and a time of 240 seconds which does not include turning adjustment time. The performance of the end-to-end autonomous vehicle is measured in proportion to the time it takes to drive in autopilot mode without human intervention. During the actual measurement, it takes 15 seconds to manually intervene the vehicle twice, so it can be calculated that the proportion of the end-to-end autonomous vehicle is:

（240-15）÷240×100% = 93.75% (3)

4. Conclusion and prospect

In this paper, an algorithm is proposed to convert the 64-line 3D Lidar point cloud data into the depth image required by the deep learning model, and the data of model input and labels are precisely synchronized. An end-to-end deep learning system based on convolutional neural network is designed. The matched data are input into the system for training to obtain an end-to-end deep learning model. Finally, the model is used to test the vehicle. The test results show that the end-to-end autonomous driving method based on 3D Lidar can effectively complete the autonomous driving task on the road in the demonstration area, which has a high value of continuing research.

Acknowledgments

Thanks to the "1612" project of Shanghai automotive industry science and technology development foundation.

5. References

[1] Nicolas Audebert, Bertrand Le Saux, Sébastien Lefèvre. *Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks [J]*. ISPRS Journal of Photogrammetry and Remote Sensing, 2017.

[2] Le Yao. *Deep Learning of Semi-supervised Process Data with Hierarchical Extreme Learning Machine and Soft Sensor Application[A]. The 28th China process control conference CPCC(2017） China association of automation process control professional committee [C]*:,2017:1.

[3] Marco Bevilacqua, Jean-François Aujol, Pierre Biasutti, Mathieu Brédif, Aurélie Bugeau. *Joint Inpainting of Depth and Reflectance with Visibility Estimation [J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2017.*

[4] Bai Chenjia. *Research on autonomous driving methods based on computer vision and deep learning [D]. Harbin Institute of Technology,2017.*

[5] Yang Q, Yang R,Davis J, Nist′er D. *Spatial-depth super resolution for range images. CVPR. 2007*

[6] Pan Yixiao. *Research on 3d modeling technology based on deep learning [D]. Central south university, 2014.*

[7] Wang Jun. *Key technology research on environment perception system of unmanned vehicle [D]. University of science and technology of China,2016.*

[8] Yu Runsheng. *Deep Reinforcement Learning Based Optimal Trajectory Tracking Control of Autonomous Underwater Vehicle[A]. Proceedings of the 36th China control conference (D) [C]. China association of automation control theory professional committee:, 2017:8.*

[9] Wang Mengwei. *Research and implementation of vehicle feature recognition system based on deep learning [D]. University of electronic science and technology, 2016.*

[10] HU Aiqin. *Deep Boltzmann Machines based Vehicle Recognition[A]. Northeastern University, IEEE Singapore industrial electronics branch, IEEE Harbin control system branch. Proceedings of the 26th China conference on control and decision-making [C]. Northeastern University, IEEE Singapore industrial electronics branch, IEEE Harbin control system branch:, 2014:6.*