Distance Measurement Method for Obstacles in front of Vehicles Based on Monocular Vision

Weiyue Gao¹, Yutuo Chen¹*, Yang Liu² and Biao Chen²

¹Department of computer and information engineering, Central South University of Forestry and Technology, Changsha, China
²Department of information engineering, Hunan Automobile Engineering Vocational College, Zhuzhou Hunan, China

*Corresponding author email: cyt28@126.com

Abstract. This paper proposes a new method based on monocular vision to detect and range obstacles in front of the vehicle for anti-collision warning during driving. Firstly, the deep learning object detection YOLOv4 algorithm is used to detect various obstacles in front of the vehicle to obtain the category and location information of the obstacles. Then an improved edge detection algorithm is used to adjust the position of the detection frame to improve the object positioning accuracy of the detection algorithm. Next, according to the camera imaging principle and geometric relationship, conversion model from the three-dimensional coordinates of the road surface to the two-dimensional coordinates of the image plane is obtained and distance measurement is performed. Finally, the cubic curve fitting of the obtained measurement data is performed, and the distance measurement process and algorithms are optimized to improve the distance measurement accuracy. The average error in the range of 50m is 0.54m, and the average error in the range of 80m is 0.78m. Through experimental analysis and comparison, the results show that the method in this paper can achieve accurate and effective monocular vision ranging.

Keywords: Monocular vision; Object detection; YOLOv4 algorithm; Obstacle ranging; Cubic Bezier curve fitting.

1. Introduction

Today, with the rapid development of intelligent transportation technology, the use of machine vision for vehicle collision prevention warning has become a hot research topic of advanced driver assistance systems in recent years. As an essential part of intelligent connected cars and advanced driving assistance systems, the detection and range-measuring of obstacles in front of vehicles are of great significance to improve the safety of automobile driving.

At the present stage, there are two main categories in the study of obstacles in front of vehicles based on machine vision: multi-ocular system and monocular ranging system. The monocular vision system uses the camera's internal parameters and geometric relations for distance measurement. Compared with the multi-ocular system, it has the advantages of a more straightforward structure, smaller calculations, and higher real-time performance. At present, research on monocular vision ranging methods can be divided into two major classes: methods based on data regression modelling and methods based on geometric positional relationship. The advantage of the data regression modelling method [1] is that it does not need to consider lens distortion and optical path deviation. This method establishes a correspondence with the image plane coordinates by obtaining a large number of sample points, but for a camera system with different parameters needs to be remodeled. The method of geometric position relationship uses auxiliary parameters such as camera placement height and pitch angle to model and
coordinate transformation. The method is more flexible, less computationally intensive, and more versatile than the data regression approach. Literature [2] used the pinhole imaging principle to propose a longitudinal vehicle distance detection method. It can achieve good results in flat and obstacle vehicle in the center of the image, but the error in detecting obstacles on both sides of the road is larger. Literature [3] used the vehicle width proportional relationship for distance measurement, which is susceptible to the influence of the diversity of vehicle width and the change of vehicle shooting angle resulting in a poor ranging effect. The method in [4] is more suitable for most of the scenes, based on the camera projection model and the geometric relationship between the pavement coordinate system with the image coordinate system.

In order to solve the problems of low localization accuracy and poor generality of monocular vehicle ranging systems, this paper proposes a distance measurement method of obstacles in front of the vehicle combining deep learning and geometric derivation. Firstly, the deep learning network YOLOv4 [5] is used as the core to construct the object detection algorithm framework, which predicts the object category and location information, namely the detection frame. In order to solve the problem of the large position error of the detection frame, an improved edge detection algorithm is used to correct it. Then, according to the camera projection model, the corresponding relationship between pavement coordinate system and image plane coordinate system is established by geometric relationship, we optimize the range measurement process and algorithm, and finally used the fitting algorithm to fit the measuring data to reduce the range measurement error.

2. YOLOv4 Object Detection Algorithm

YOLOv4 algorithm is the fourth generation of YOLO (You Only Look Once) series object detection algorithm. Based on the YOLOv3 [6] algorithm, the YOLOv4 algorithm further strengthens the detection accuracy and detection speed. The algorithm introduces CSPNet [7] based on the Darknet-53 network, changes the backbone network framework to CSPDarknet-53 and introduces the Leaky-ReLU activation function to improve the learning capability of the network effectively. FPN [8] (Feature Pyramid Networks) is used for down-sampling to extract robust semantic features for object classification, and PAN [9] (Path Aggregation Network) is used to up-sampling to extract localization features for location detection, which is significantly improves the performance of the model in object detection. To ensure driving safety, the accuracy and real-time performance of the detection algorithm are crucial. Compared with other object detection algorithms, the YOLOv4 algorithm has an excellent average detection accuracy and speed while ensuring real-time performance. Compared with the YOLOv3 algorithm, the average detection accuracy AP (Average Precision) is increased by 10%, and the FPS (frame rate per second) is increased by 12%. It can meet the real-time detection requirements for obstacles in front of the vehicle, such as motor vehicles, non-motor vehicles, pedestrians, roadblocks, etc. during driving. Input the image collected by the camera into the trained YOLOv4 algorithm. Figure 1 shows the results of object detection.

![Figure 1. Obstacle Object Detection Results](image)

3. Correction of Object Detection Frame Position

Through the analysis of a large number of YOLOv4 algorithm detection results, it is found that the vertical position deviation is usually within 30 pixels. Take the midpoint of the bottom edge of the detection frame as the centre, take 50 pixels for the upper, lower, left, and right sides to intercept the
image, take a 100×100 pixel picture and perform edge detection operations on this picture. Canny edge detection operators are improved by the mean filter and Median filter to enhance edge detection noise resistance, continuity and positioning accuracy, and reduce miscalculation rates. Edge detection results for objects using improved Canny edge detection operators are shown in Figure 2.

![Figure 2. Improved edge detection results for the Canny edge detection operators](image)

(a) Grayscale images  (b) Edge Detection

Results

(a) Before adjusting  (b) After adjusting

Figure 3. Detection frame adjustment results

The horizontal line below the object in Figure 3 is the bottom edge of the detection frame and the midpoint is marked. First, the average value of the vertical coordinates of all pixels on the actual bottom edge of the object in edge detection is calculated, and then the vertical correction distance of the midpoint of the bottom edge of the frame is obtained by making a difference between this value and the vertical coordinates of the pixels at the midpoint of the actual bottom edge. The corrected result is shown in Fig. 3(b).

4. Obstacle Ranging Algorithm

The image acquisition process converts the objective world's three-dimensional road coordinates system into the camera's two-dimensional camera coordinate system. In contrast, the ranging process is the inverse of image acquisition. We first need to get the coordinates of the object on the camera plane, and then use the camera's horizontal view, vertical view, elevation angle, placement height and clarity and other parameters to convert the camera plane coordinates into three-dimensional road coordinates. Then the distance between the obstacle and the camera is calculated by the coordinates.

The world three-dimensional road surface coordinate system is shown in Figure 4(a). Plane ABK represents the road surface, ABCD is the area where the road surface is photographed by the camera lens O, the intersection of the camera's optical axis with the road surface is G, and J is the vertical of the camera on the road surface. OJ is the mounting height of the camera h. \( \angle AOB \) and \( \angle COD \) are the horizontal angles of view of the camera, the size is \( 2\beta \), \( \angle IOL \) is the vertical angle of view of the camera, the size is \( 2\alpha \), and \( \angle JOG \) is the pitch angle of the camera, the size is \( \gamma \).

![Figure 4. The three-dimensional road and camera plane coordinates, and camera imaging model](image)

(a)  (b)  (c)

The image plane coordinate system is shown in Figure 4(b), and the range of the picture taken by the camera is rectangle abcd. \( H \) and \( W \) are the height and width of the image respectively. Figure 4(c) shows the Y-axis camera imaging model. The angle between the object and the optical axis OG. \( \angle GOP_y \) is \( \theta_0 \). According to the camera imaging principle, \( p_y \) is the projection of point P on the Y-axis, \( p_{y,g} \) is the Y-axis coordinate value \( Y_P \) of the point P on the image plane, and the corresponding point \( P(X_P,Y_P) \) on the image plane is \( p(x_P,y_P) \) in the surface coordinate system. The mapping of pavement coordinates to the image coordinate system can be derived from the geometric relationship as shown in equations (1).
\[ Y_p = 2h \times y_p \times \tan \alpha \left[ \frac{1 + \tan^2 \gamma}{H - 2y_p \cdot \tan \gamma \cdot \tan \alpha} \right] \]
\[ X_p = \frac{2x_p \times h \left( KG + y_p \tan \beta \right)}{W \times KG \times \cos \gamma} \]
\[ KG = \frac{h \left[ \tan \gamma - \tan \left( \gamma - \alpha \right) \right] \times \cos \left( \gamma - \alpha \right)}{\cos \left( \gamma - \alpha \right) - \cos \gamma} \]  

Where \( W \) is the width of the image, \( H \) is the height of the image, \( h \) is the height of the camera, \( 2\beta, 2\alpha, \) and \( \gamma \) are the horizontal angle of view of the camera, the vertical angle of view of the camera, and the pitch angle of the camera.

Calculate the coordinates \((x_1, y_1)\) of the midpoint of the bottom edge of the object on the image. Used equation (1) and (2) to convert it to coordinates \((X_1, Y_1)\) in the world coordinate system. The distance between the obstacle ahead and the vehicle is:

\[ d = JP = \sqrt{(h \times \tan \gamma + Y_1)^2 + X_1^2} \]  

5. Cubic Curve Fitting of Measurement Data

To further improve the accuracy of the distance measurement, this paper fits the actual measured distance to the predicted distance corrected by the algorithm using the cubic Bezier curve equation. It interpolates the data to reduce the distance measurement error.

The cubic Bezier curve equation is:

\[ Q(t,V) = \sum_{i=0}^{3} B_i(t)V_i \]

Where \( B_i(t) = C_i^3 t^i \left( 1-t \right)^{3-i}, \quad t \in [0,1] \); \( V_k \) is the control point \((k=0, 1, 2, 3)\), and \( P_{d(x_0, y_0)} \) to \( P_{d(x_n, y_n)} \) are the data measurement through the curve According to the properties of cubic Bezier curve: \( V_0 = P_0 \), \( V_3 = P_n \).

The offset from the data point to the curve is:

\[ d_i = \left[ Q(t,V) - P_i \right] \cdot \left[ Q(t,V) - P_i \right] \]

The total offset from the data point to the curve is:

\[ D(t,V) = \sum_{i=1}^{n-1} d_i = \sum_{i=1}^{n-1} \left[ Q(t,V) - P_i \right] \cdot \left[ Q(t,V) - P_i \right] \]

To minimize \( D(t, V) \) (sum of the distance from the data point to the curve), the two intermediate control points \( V_1 \) and \( V_2 \) of the cubic Bezier curve need to be solved to construct the equation. The derivation process is as follows:

Determine the partial derivatives of \( V_1 \) and \( V_2 \) for \( D(t, V) \) and make them equal to 0:

\[ \frac{\partial D(t,V)}{\partial V_1} = 2 \sum_{i=1}^{n-1} \frac{Q(t,V)}{\partial V_1} \left[ Q(t,V) - P_i \right] = 0 \]
\[ \frac{\partial D(t,V)}{\partial V_2} = 2 \sum_{i=1}^{n-1} \frac{Q(t,V)}{\partial V_2} \left[ Q(t,V) - P_i \right] = 0 \]  

\[ \Rightarrow \begin{align*}
\sum_{i=1}^{n-1} \left[ \frac{B_i(t)}{P_1} \right]^2 & \sum_{i=1}^{n-1} B_i(t) B_j(t) = \sum_{i=1}^{n-1} B_i(t) \left[ P_1 - B_i(t) V_0 - B_j(t) V_j \right] \\
\sum_{i=1}^{n-1} B_i(t) B_j(t) & \sum_{i=1}^{n-1} \left[ B_i(t) \right]^2 = \sum_{i=1}^{n-1} B_i(t) \left[ P_1 - B_i(t) V_0 - B_j(t) V_j \right]
\end{align*} \]  

\[ \Rightarrow \begin{align*}
\sum_{i=1}^{n-1} \left[ B_i(t) \right]^2 & \sum_{i=1}^{n-1} B_i(t) B_j(t) = \sum_{i=1}^{n-1} B_i(t) \left[ P_1 - B_i(t) V_0 - B_j(t) V_j \right] \\
\sum_{i=1}^{n-1} B_i(t) B_j(t) & \sum_{i=1}^{n-1} \left[ B_i(t) \right]^2 = \sum_{i=1}^{n-1} B_i(t) \left[ P_1 - B_i(t) V_0 - B_j(t) V_j \right]
\end{align*} \]
The control points $V_1$ and $V_2$ are obtained by solving equation (8), which is the cubic Bezier curve equation of equation (4).

6. Results
In this paper, a wide-angle camera with $4,032 \times 3,024$ image pixels, 4.216mm focal length, f/1.8 aperture, and $65^\circ$ horizontal viewing angles and $59.6^\circ$ vertical viewing angles is used. The program running environment is Python 3.6, the YOLOv4 object detection algorithm is implemented based on the deep learning framework Tensorflow, the version number is 1.11, and the parallel computing library is CUDA Toolkit 9+cuDNN 7.1. The partial results after adjusting the detection frame position are shown in Figure 5.

![Figure 5. Distance measurement results after adjusting the detection frame position.](image)

![Figure 6. Fitting Results](image)

We placed obstacle cars on the road at a distance of 5 to 80m from the camera, and measure the distance every 5m. The experimental results are shown in Table 1. The average error of the detection frame before adjustment is 6.21m, and the average error after adjustment is 1.504m. The 60 sets of measured distances and corrected measured distances are substituted into Eq. (8) to solve for control points $V_1$ and $V_2$ by the cubic Bessel curve fitting algorithm in Section 5. Combining control points $V_0=P_0$, $V_3=P_n$ enter into equation (4), and expanding the cubic Bezier curve parameter equation is obtained as follows equation (9):

\[
\begin{align*}
  x &= 5.1770(1-t)^3 + 31.973t(1-t)^2 + 55.5767(1-t)t^2 + 85.9256t^3 \\
  y &= 5(1-t)^3 + 31.1697t(1-t)^2 + 58.5577(1-t)t^2 + 80t^3
\end{align*}
\] (9)

Enter the corrected measurement distance into equation (9) to get the fitted distance, and the measured distance after curve fitting is shown in the fourth column of Table 1. The start and end errors according to the Bezier curve property are zero. The average error distance after the correction of the detection box position is 1.504m, and the average error distance after the cubic Bezier curve fits is 0.776m, which significantly reduces the range error by fitting.

| Actual Distance(m) | Initial measured distance (m) | Measured distance after position correction (m) | Measured distance after curve fitting (m) | Measured distance error(m) | Measured distance error (%) |
|-------------------|-----------------------------|-----------------------------------------------|------------------------------------------|---------------------------|----------------------------|
| 5                 | 5.3091                      | 5.17 0                                        | 5                                         | 0                         | 0                          |
| 10                | 9.0496                      | 9.4581                                        | 9.2509                                    | 0.7491                    | 7.491                      |
| 15                | 14.2977                     | 14.9761                                       | 14.7970                                   | 0.2030                    | 1.355                      |
| 20                | 21.2773                     | 20.6372                                       | 20.5371                                   | 0.5371                    | 2.685                      |
| 25                | 25.9248                     | 25.5887                                       | 25.5730                                   | 0.5730                    | 2.292                      |
| 30                | 28.5890                     | 29.3857                                       | 29.4290                                   | 0.5710                    | 1.903                      |
| 35                | 41.6549                     | 35.9956                                       | 36.0952                                   | 1.0952                    | 3.129                      |
| 40                | 48.0998                     | 40.7217                                       | 40.7996                                   | 0.7996                    | 1.999                      |
| 45                | 50.0178                     | 44.7131                                       | 44.7160                                   | 0.2840                    | 0.631                      |
| 50                | 48.5291                     | 50.7808                                       | 50.5448                                   | 0.5448                    | 0.109                      |
| 55                | 61.6844                     | 53.3838                                       | 52.9917                                   | 2.0083                    | 3.651                      |
| 60                | 78.9836                     | 62.3644                                       | 61.1455                                   | 1.1455                    | 1.909                      |
| 70                | 93.6298                     | 75.7761                                       | 72.3620                                   | 2.3620                    | 3.374                      |
| 80                | 90.7885                     | 85.9256                                       | 80                                         | 0                         | 0                          |
Figure 6 shows the results of the cubic Bezier curve fits, where the dots are the corrected measured distance data points and the triangles are the curve control points. The experimental results show that the average error distance is 0.5356m and the maximum error is 1.0952m at actual distances of 5-50m. When the actual distance is 55m, the maximum distance measurement error is 2.0083m. The average error of the measured distance after fitting is 0.7766m.

7. Conclusion
This paper presents a method for measuring the distance between a moving vehicle and an obstacle ahead, based on a monocular vision system. The deep learning-based object detection YOLOv4 algorithm is used to detect obstacles in front of the vehicle, solving the problem of poor generalizability of traditional obstacle detection algorithms. In this paper, an improved Canny edge detection operator is used to solve the problem of inaccurate prediction of object position information by the object detection algorithm. The object position is adjusted by calculating the difference of vertical coordinates of the bottom edge position of the detected frame. Based on the camera projection model, the range model is established by geometric derivation, and the range data is fitted with cubic Bezier curves. The results shown that the method in this paper greatly improves the accuracy of obstacle distance measurement. The bumps suffered during the driving process will have a large error on the algorithm of this paper, we will conduct further research to solve the problem.

Acknowledgments
This work is supported by Scientific Research Project of Hunan Education Department (18B566) and National Natural Science Foundation of China (11771453)

References
[1] YANG P. Forward vehicle detection and ranging algorithms based on deep learning. XiDian University.
[2] YANG W, WEI L, GONG J Q, et al. (2012) Research on Detection of Longitudinal Vehicle Spacing Based on Monocular Vision. Computer Measurement & Control, 20(08), pp.2039-2041, 2044.
[3] Ki-Yeong P, Sun-Young H. (2014). Robust Range Estimation with a Monocular Camera for Vision-Based Forward Collision Warning System. The Scientific World Journal, (2014-12-9), 2014, 2014:923632.
[4] GUO L, XI Y C, LI K Q, et al. (2006). Study on Real-time Distance Detection Based on Monocular Vision Technique. Journal of Image and Graphics, pp.74-81.
[5] Bochkovskiy A, Wang C Y, Liao H Y M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv: 2004.10934v1.
[6] REDMON J, FARHADI A. (2018). YOLOv3: an incremental improvement. arXiv: 1804.02767v1.
[7] Wang C Y, Liao H Y M, Yeh I H, et al. (2019). CSPNet: A New Backbone that can Enhance Learning Capability of CNN. arXiv: 1911.11929v1.
[8] LIN T Y, DOLLAR P, GIRSHICK R, et al. (2017). Feature pyramid networks for object detection/2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017. Honolulu, HI, IEEE, pp.936-944.
[9] Liu S, Qi L, Qin H, et al. (2018). Path Aggregation Network for Instance Segmentation. arXiv: 1803.01534v4.