A Real-time Visual Tracking and Distance Measuring Algorithm based on SSD and Data Regression

Tongpo Zhang[1], Zejian Kong[1], Tiantian Guo[1],Miguel Lopez-Benitez[3,4], Enggee Lim[1], Fei Ma[2], Limin Yu[1]
1.School of advanced technology, Xi'an Jiaotong-Liverpool University,Suzhou, China
2.School of science, Xi'an Jiaotong-Liverpool University, Suzhou, China
3.Electrical Engineering and Electronics, University of Liverpool, Liverpool, United Kingdom
4.ARIES Research Centre, Antonio de Nebrija University, 28040 Madrid, Spain

***Abstract*—Real-time visual tracking and distance measuring algorithms are of critical importance in industry areas. In this paper, we propose a novel solution for this practical application problem with precision, and test the performance of the algorithm with experiments. First, we compare the ability of the traditional visual tracking algorithms with deep learning visual tracking methods. Three deep learning visual tracking methods are compared and evaluated to find the most suitable method that can meet the real-time requirement of target tracking. We then create a real-time visual tracking and distance measuring algorithm based on Single Shot Multibox Detection (SSD). Two methods are proposed for the distance measurement algorithm, and both methods are tested. The experiments are implemented to validate the performance of the designed approach. The experimental results demonstrated the effectiveness and precision of the designed real-time visual tracking and distance measuring algorithm.**

Keywords—SSD algorithm, object detection, real-time distance measurement

I. INTRODUCTION

Research on visual tracking and distance measurement in real-time has continued for a long time. The goal of this function is to implement a better and wiser application in other areas to meet the relative requirement such as helping the navigation system to get more position information, making the robot be able to follow the moving object or asking for an analysis of the targets in a video.

There are two major components in the process of real-time visual tracking and distance measurement which include object detection and tracking, distance measuring. Firstly, the tracking target is selected. Then the vision tracking algorithm works and enables a rectangular box to lock the target no matter how the target moves in view. In addition, the rectangular box can change its size according to the size of the target in the view of the screen and re-detect the target if the target is obscured or moves out of the view. Meanwhile, the distance measuring algorithms is able to measure the distance between the camera and the target based on the position of the earth point of the target. While a series of tracking algorithms have been proposed in the past, it remains challenging that traditional tracking algorithms can not address the problems of the deformation, occlusion of targets [1-4,5,7,11]. This article proposes a real-time visual tracking and distance measuring algorithm based on Single Shot Multibox Detection (SSD) and data regression to solve the problem.

The remainder of this paper is organized as follows: In Section 2, we formulate a comparison of the traditional visual tracking method and deep learning visual tracking method. In section 3, we pursue the structure of the created algorithm. Experiments and conclusions are respectively presented in section 4 and section 5.

II. RELATED WORK

In this section, we recall the traditional visual tracking methods and the deep learning tracking methods.

Four traditional visual tracking methods are tested which are MIL, KCF, TLD and MOSSE. Multiple Instance Learning (MIL) can avoid the problem of slight inaccuracies in the tracker leading to incorrectly labeled training examples, which degrades the classifier and can cause further drift. It was created by Boris Babenko's team in 2009 which can lead to a more robust tracker with fewer parameter tweaks [1]. Joao F. Henriques's team derived a kernelized correlation filter (KCF) with a fast multi-channel extension of linear correlation filters, via a linear kernel, which was called dual correlation filter (DCF) in 2015 [2].

Tracking-Learning-Detection (TLD) is a tracking framework that explicitly decomposes the long-term tracking task into tracking, learning, and detection which was proposed by Zdenek Kalal's team. It shows a significant improvement over state-of-the-art approaches [3]. Minimum Output Sum of Squared Error (MOSSE) is a type of correlation filter created by David S. Bolme's team to produce stable correlation filters when initialized using a single frame [4].

To evaluate the performances of traditional tracking algorithms, the criteria of Speed, Re-detection and Frame anchor adaptation are created. Table 1 shows the comparisons of monocular tracking algorithms.

TABLE I. COMPARISONS OF MONOCULAR TRACKING ALGORITHM

Name	Speed(fps)	Re-detection	Frame anchor adaptation
MIL	15	No	No
KCF	220	Good for shielding Bad for out of view	No
TLD	16	Good for shielding and out of view	Yes
MOSSE	1962	No	No

These tracking algorithms are run in real-time on the personal computer whose GPU type is NVIDIA Quadro P600. The speed means the number of frames that the algorithm can track the target in one second. Different projects have different requirements for the timeliness of the algorithms. From the table, MOSSE has the highest speed which is 1962 fps. The re-detection evaluates the performances of the tracking algorithm to detect the target again when the target appears in the view integrally after being occluded or moving out of the view. The result is that MIL does not have re-detection ability. KCF can re-detect the target when the target is occluded but has poor performances when the target moves out of the view completely. TLD has the strongest re-detection ability. MOSSE does not have the ability to re-detect. Frame anchor adaptation means the algorithm can track the target successfully when the size of the target in the view changes continuously. From the table, it can be seen that the frame anchors of MIL, KCF, MOSSE cannot adapt to the size of the target while TLD can.

Three deep learning tracking models are tested which are faster R-CNN, YOLO V5 and SSD. After the accumulation of r-cnn and fast R-CNN (Region-based Convolution Neural Networks), Ross B. girshick created a new fast RCNN, called faster R-CNN in 2016. Structurally, faster R-CNN has integrated feature extraction, proposal extraction, bounding box region and classification into one network. It greatly improves comprehensive performance, especially in terms of detection speed. It is a single, unified network for object detection [5-6]. YOLO, called You Only Look Once is an approach to object detection. It was raised by Joseph Redmon's team in 2015 [7]. In 2020, the Ultralytics raised YOLOV5, there is no research article now. However, based on the current displayed data and some other researchers' testing [8-10]. YOLOV5 can be regarded as a useful object detection method. It is still being updated by the Ultralytics. Single Shot MultiBox Detector (SSD) is a method for detecting objects in images using a single deep neural network raised by Wei Liu's team in 2016 [11]. SSD has the advantage of high speed because the object classification and predict anchor regression are done simultaneously. SSD utilizes CNN to extract features and sampling from the feature map at different locations with different scales densely and evenly. Table 2 shows the performance of traditional monocular tracking algorithms and deep learning tracking algorithms tracking the object in a video with 936 frames.

TABLE II. TRACKING TESTING RESULTS

Algorithm	Tracking success(frames)
MIL	179
KCF	445
TLD	74
MOSSE	229
Yolo	936
Faster-rcnn	936
SSD	936

It can be seen from table 2 that traditional tracking algorithms are easier to cause tracking failure compared with deep learning tracking algorithms. The traditional tracking algorithms have high processing speed, but the initial frame needs to be framed manually or with the help of a target detection algorithm. In addition, when the angle of view of the target changes, such as the side view in the field of view after the object turns, the traditional algorithm with the tail of the object as the initial frame will lose the target or drift. Furthermore, traditional algorithms are easier to lose targets when doing long time tracking, the anchor frame will be no longer locked to the target. The deep learning tracking algorithms do not need an initial frame framed manually or with the help of another target detection algorithm. They can identify the target from all angles. Therefore, it can be concluded that the deep learning tracking algorithms have better robust. Moreover, according to the testing results, the SSD model is faster than the Yolo V5 model and Faster-rcnn model which can meet the need of real-time detection and distance measuring. That is the reason why the SSD model was used.

III. PROPOSED METHOD

Our approach is based on the combination of the SSD model and Data regression model. Figure 1: The flowchart of the proposed method shows the progress of the designed method. First, the model of SSD is loaded. The program will then read the frame to detect the target. In addition, a default box will lock the target and SSD outputs the coordinate of the center position of the lower border of the default box. After that, the distance between the camera and the camera will be calculated based on the output coordinate of SSD. The result will be shown on the screen. Subsequently, the next frame will be read until all frames have been read. If there is no target in the frame, the algorithm will continue to detect but will neither lock anything nor output any distance information. For real-time detection, the algorithm reads the real-time images from the camera. The program will not stop until receives a stopping command.

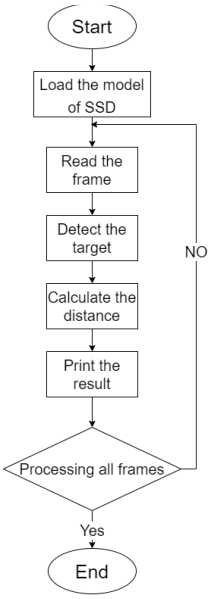

Figure 1: The flowchart of the proposed method

The first distance measuring method is based on imaging geometry which can be seen in functions 1 to 5.

$$Z_c\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} R & T \end{pmatrix}\begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} = M_1 M_2 \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \quad (1)$$

(u,v,1) is a vector that presents the position of the target in the pixel coordinate system. f_X, f_Y are the Focal lengths of the camera in x-direction and y-direction respectively. R is the Rotation matrix and T is the Translation matrix. (X_W, Y_W, Z_W,1) represents the coordinate of the target in the world coordinate system. M_1 is the intrinsic matrix and M_2 is the extrinsic matrix. The formula(1) describes how the imaging system converts the position of the target in the real world to the imaging plane. By measuring the position of the target in the real world and outputting the coordinate of the target from the software, the world coordinates and pixel coordinates are easy to acquire. But other parameters cannot be acquired directly.

Because different cameras have different converting parameters while users do not know these parameters in advance, Zhang Zhenyou method can be used to calibrate the intrinsic matrix of the camera. It is a flexible technique to easily calibrate a camera [12]. By taking photos of a standard checkerboard of black and white rectangles from dozens of different angles, the method can resolve the equations about the intrinsic matrix of the camera. Considering the fact that the target moves relative to the camera, the extrinsic matrix always changes. Therefore, though ZhangZhenyou method can provide the extrinsic matrix of each calibrating image, these matrixes are useless for other situations. Therefore, the geometry model is created to compensate for the unknown extrinsic matrix with the height of the camera relative to the ground, the angle between the optical axis of the camera and the horizontal line, the camera and the target moving in the flat ground as prior conditions. It is based on a Vision-based Adaptive Cruise Control (ACC) system [13].

$$Z_C = \cos\left(\arctan\left(\frac{v-v_0}{f_y}\right)\right) \times \frac{H}{\sin\left(\alpha+\arctan\left(\frac{v-v_0}{f_y}\right)\right)} \quad (2)$$

$$X_C = Z_C \frac{u-u_0}{f_x}, \quad (3) \quad Y_C = Z_C \frac{v-v_0}{f_y}, \quad (4)$$

$$D = \sqrt{X_C^2 + Y_C^2 + Z_C^2} \quad (5)$$

X_C, Y_C, Z_C: Three coordinates in Camera coordinate system H: The height of the camera relative to the ground，α：The angle between the optical axis of the camera and the horizontal line, D: the distance between the target and the camera.

The tracking algorithm outputs the coordinates of the target in the Pixel coordinate system: u and v. The Pixel coordinates should be converted into Image coordinates and then to Camera coordinates. Formula 1 uses prior conditions to calculate the depth value(Z_C) based on geometry relations. Formula 2 and Formula 3 use the parameters of intrinsic matrix and depth value to calculate the other two coordinates in the camera coordinate system. Finally, Formula 4 calculates the distance between the target and the camera.

The second distance measurement algorithm is based on Data regression modeling. To simplify the model, the method only focuses on the distance when the target locates right ahead of the camera. There is a mapping relation between the distance and the Pixel y-coordinate: D=f(y). By analyzing, the fitting of the mapping relation can be reduced to a multi-variable non-linear fitting problem:

$$D = c_0 + c_1 y + c_2 y^2 + \ldots\ldots + c_m y^m. \quad (6)$$

Firstly, the data pairs of real distances and values of pixel coordinates are collected. Then the least square method is used and functions of different degrees are selected to fit these data points. Finally, the curves are evaluated and the function which has the best fitting curve is determined.

IV. EXPERIMENT

A. Experiment setup

The hardware used contains the Dell personal computer with NVIDIA Quadro P600 GPU. The camera is the front-facing camera of the computer which is an HP Wide Vision HD Camera made by Microsoft. Operating workspace environment contains windows 10, cuda 11.4，cudnn7.6.5, pycharm and pytorch. There is a small tracking target and the experiments are conducted indoors. To train the models of deep learning visual tracking algorithms, a data set consisting of 300 images

is created. Firstly, we take 300 photos of the target from different angles and distances indoor environment with multiple objects as backgrounds. Then the photos are used to create a data set with VOC format. The SSD model uses the data set to train its model and load the model to predict. The target is laid on the ground which is in the front-facing of the camera and then we run the program. To test the performance of the algorithm, the target moves from near to far. Meanwhile, we record the output distance value and use a tapeline to measure the real distances. Then we compare the measuring distance with the real distance.

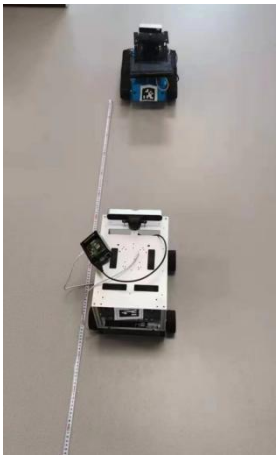

Figure 2: Distance Measurement Experiment

B. Experiment results

Figure 3 shows the fitted curve for the processed pixel coordinates and real distance of the experiment. X-axis represents the processed pixel value of the center position of the lower border of the default box in the pixel coordinates while the Y-axis represents the real distance between the camera and the target. The blue point is the collected data for the modeling. The red curve is the fitted curve for the function, which is the result of data regression modeling.

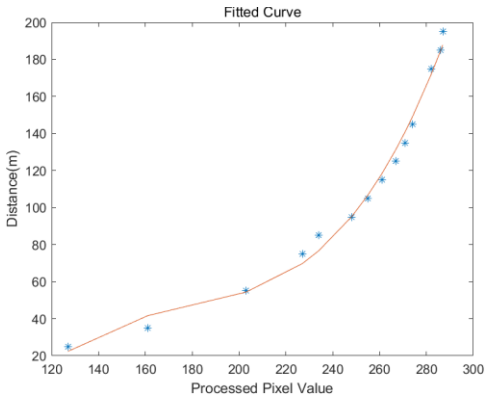

Figure 3: The fitted curve for the processed pixel coordinates and real distance

The image has 480 pixels in y-coordinate in total. The original point locates in the top left corner of the image in the pixel coordinate system. That means the value of the y-coordinate increases from the top to the bottom of the image. However, when the target appears at the bottom of the image, it means a shorter distance. To create the model which is in line with the intuitive feeling, the mapping function becomes D=f(480-y)=f(h), h:pixel height. 14 pairs of data are collected. The result shows that the quadratic function can not meet the requirement of monotone increasing in the interval and the quartic function are over-fitting. So the cubic function has the best effect. The fitting result can be seen in Table 3: The fitting results. Algorithm 1 is a distance measuring method based on imaging geometry. Algorithm 2 is a distance measuring method based on Data regression.

TABLE III. THE FITTING RESULTS

	Algorithm 1		Algorithm 2	
Real distances(m)	Measuring distance(m)	Error	Measuring distance(m)	Error
0.25	0.11	56%	0.21	16%
0.3	0.22	26.7%	0.32	6.7%
0.4	0.3	25%	0.45	12.5%
0.5	0.39	22%	0.49	2%
0.6	0.47	21.7%	0.59	1.7%
0.7	0.57	18.5%	0.63	10%
0.8	0.66	17.5%	0.73	8.8%
0.9	0.78	13%	0.88	2.2%
1.0	0.92	8%	0.98	2%
1.1	1.07	2.7%	1.08	1.8%
1.2	1.18	1.6%	1.17	2.5%
1.3	1.26	3%	1.32	1.5%
1.4	1.37	2%	1.39	0.71%
1.5	1.59	6%	1.49	0.67%
1.6	1.79	11%	1.602	0.13%
1.7	1.88	10%	1.69	0.59%
1.8	2.15	19%	1.78	1.1%
1.9	2.76	45%	1.82	4.2%
2.0	3.34	67%	1.85	7.5%

Based on table 3, algorithm 1 has around 10% percent error when the real distance ranges from 1m to 1.6m. When the distance is smaller than 0.6m and larger than 2.15m, the error is larger than 20%. Because as the distance increases, one pixel corresponds to a larger distance in the real world, the error is large when the distance is far. When the distance is small, the error resulting from camera calibration is enlarged. As a result, the distance measurement algorithm can only have good performance in a particular distance. Algorithm 2 has much smaller errors than algorithm 1. Its error rate is smaller than 12%. However, if the real distance is lower than 0.24m, the algorithm will output negative values, which is impractical. As a result, though the algorithm has a higher degree of accuracy, it also has some limitations.

V. Conclusion

In this article, a novel method for real-time visual tracking and distance measurement based on SSD and data regression is proposed. This algorithm can measure the distance of the tracking object with high accuracy in real-time. The error can be controlled at a centimeter-level. It can meet relevant industrial needs to a certain extent. The algorithm architecture and design framework facilitate further improvement on the tracking system. Better computation power of the hardware improves the system performance. The SSD algorithms can also be improved by adding anti-convolution layers to optimize the performance to detect small objects and increase its semantic comprehension ability. The distance measurement algorithms can be replaced by the monocular deep estimation algorithms to handle experimental environment that is not be limited to a flat ground.

Acknowledgment

This research was partially funded by the Research Enhancement Fund of XJTLU (REF-19-01-04), National Natural Science Foundation of China (NSFC) (Grant No. 61501380),and by AI University Research Center (AI-URC) and XJTLU Laboratory for Intelligent Computation and Financial Technology through XJTLU Key Programme Special Fund (KSF-P-02), and Jiangsu Data Science and Cognitive Computational Engineering Research Centre..

References

[1] Babenko B , Yang M H , Belongie S . Visual tracking with online Multiple Instance Learning[C]// Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009.

[2] J.F. Henriques, R. Caseiro, P. Martins and J. Batista, "High-Speed Tracking with Kernelized Correlation Filters," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 3, pp. 583-596, 1 March 2015, doi: 10.1109/TPAMI.2014.2345390.

[3] Z. Kalal, K. Mikolajczyk and J. Matas, "Tracking-Learning-Detection," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 7, pp. 1409-1422, July 2012, doi: 10.1109/TPAMI.2011.239..

[4] D. S. Bolme, J. R. Beveridge, B. A. Draper and Y. M. Lui, "Visual object tracking using adaptive correlation filters," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 2544-2550, doi: 10.1109/CVPR.2010.5539960.

[5] Ross B. Girshick,Dectinc Chen, LoneStar, Max Ehrlich,23 Jan 2018, 2022 GitHub, Inc. Available online: https://github.com/rbgirshick/py-faster-rcnn.

[6] Ren S , He K , Girshick R , et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, 39(6):1137-1149.

[7] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

[8] Joseph Nelson, Jacob Solawetz ,JUN 10, 2020, YOLOv5 is Here: State-of-the-Art Object Detection at 140 FPS, Available online: https://blog.roboflow.com/yolov5-is-here/.

[9] Glenn Jocher, ultralytics,12 Oct 2021, 2022 GitHub, Inc. Available online: https://github.com/ultralytics/yolov5.

[10] Jubayer M F , Soeb M , Paul M K , et al. Mold Detection on Food Surfaces Using YOLOv5. 2021.

[11] Liu W , Anguelov D , Erhan D , et al. SSD: Single Shot MultiBox Detector[C]// European Conference on Computer Vision. Springer, Cham, 2016.

[12] Zhang Z . A Flexible New Technique for Camera Calibration[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(11):1330-1334.

[13] G. P. Stein, O. Mano and A. Shashua, "Vision-based ACC with a single camera: bounds on range and range rate accuracy," IEEE IV2003 Intelligent Vehicles Symposium. Proceedings (Cat. No.03TH8683), 2003, pp. 120-125, doi: 10.1109/IVS.2003.1212895.