Using Efficient IoU loss function in PointPillars Network For Detecting 3D Object

Sazan Mohammed[1,3], Mohd Zulhakimi Ab Razak[1], Abdul Hadi Abd Rahman[2]
[1]Institute of Microengineering and Nanoelectronics (IMEN), Universiti Kebangsaan Malaysia, Malaysia
[2]Center for Artificial Intelligence Technology, Universiti Kebangsaan Malaysia
[3]Department of Automotive Technology, Erbil Technology College, Erbil Polytechnic University, Erbil, Iraq
Email: p103643@siswa.ukm.edu.my, zul.hakimi@ukm.edu.my, abdulhadi@ukm.edu.my

***Abstract*—Detecting three-dimensional (3D) objects has attracted the growing attention in 3D computer vision research. However, the low precision value is a significant trouble in many applications, like automatic driving, robotics, and medical applications. To solve the low precision problem, we use the 3D EIoU loss as localization loss, which emphasizes on the overlapping degree, central position, and structural shape between two rectangular bounding boxes. Furthermore, we propose an EIoU-NMS to enhance the process of suppressing redundant detecting boxes. By incorporating the 3D EIoU loss and EIoU-NMS into the PointPillars one-stage detectors, the detection performance for 3D point cloud objects is considerably improved. By using the KITTI benchmark, empirical experiments have been conducted to measure the average precision (AP) values for detecting Car, Cyclist, and Pedestrian objects.**

Keywords— object detection, localization accuracy, 3D bounding box regression, 3D loss function

I. INTRODUCTION

3D LIDARs are widely used sensors for 3D object detecting. The captured point cloud data are sparse and unstructured, which makes an urgent problem in the object detecting process. Deep learning object detection techniques are used in autonomous driving, mobile robot, and others many because of their high detection accuracy. Many studies use RGB cameras for detecting 3D objects [1][2]. During the projection process of the 3D environment to a 2D image, the spatial information will be lost. This information is very important in many applications such as path planning and decision making. On the other hand, the point cloud data contains structural and 3D spatial information about a particular area. The acquisition of point cloud data is more suitable nowadays with the fast development of LIDAR. As a result, the point cloud 3D object detection techniques have become an essential part of different 3d applications.

Low detection precision for point cloud objects is the main problem in many deep learning 3D object detection algorithms. Some technologies [3][4], use images for 2D detection algorithm, for achieving 3D object detection a bounding box regression was used. The KITTI dataset in the work [5] has achieved good results by using a perfect 2D detector for images. But these techniques have expensive time costs, and they are extremely dependent on 2D image technology. For solving these shortcomings, we propose that just the point cloud data be utilized for 3D object detection to decrease time cost, by using the new 3D EIoU loss function the alignment of 3D ground truth and the prediction bounding box can be reduced. Adding the New EIoU-NMS process to PointPillars network, enhance the precision of the 3D object detecting method.

Object classification and localization are the main tasks in any object detection process. The bounding box regression step shows an essential part in the object detection process. Numerous studies like [6][7] depend on the bounding box regression to find the object location accurately. Using a reasonable regression loss function improves the accuracy value of the bounding box and optimizes the architecture of the deep neural network. As a result, several regression loss equations have been proposed, which are also used in removing the duplicated bounding box for the Non-Maximum Suppression (NMS) process.

The popular loss functions, the l1-smooth, and l2-norm or mean square error are used mainly to optimize the bounding box. These functions cannot consider the Intersection Over Union value [8]. Furthermore, the IoU loss function has a problem and cannot be used for the evaluation process when the two bounding boxes are totally not overlapped. The Generalized Intersection over Union (GIoU) is also not convent when its value is equal to IoU [8]. The Distance Intersection over Union (DIoU) and Complete Intersection over Union (CIoU) are useful loss functions, but considering the distance between bounding boxes is not sufficient without considering the aspect ratio of the bounding boxes [9]. So adding the properties of the DIoU to CIoU will lead to a better evaluation function and this function named, the Efficient Intersection over Union (EIoU) loss function [10], which is used for the 2D object detecting process before this work and we propose a 3D EIoU loss function for increasing the accuracy of 3D object detecting values.

In this study, we propose three 3D loss functions: Complete-IoU (CIoU), Distance-IoU (DIoU), and Efficient-IoU (EIoU). We simply change the previous 2D CIoU, DIoU, and EIoU to 3D loss functions by using the 3D coordinate x, y, and z with also the three sides' terms of the bounding box, width, height, and length. The new 3D EIoU loss function leads to much faster convergence than other 3D loss functions, CIoU, DIoU, GIoU and IoU. Furthermore, we suggest that a good 3D loss for bounding box regression should consider three geometric sides' changes with midpoint distance. By using the three sides' changes alone, we further propose a 3D Complete-IoU (CIoU) loss for bounding box regression, leading to faster convergence and better performance than IoU, GIoU and DIoU losses. The 3D DIoU loss function performs better than other functions (i.e. GIoU and IoU). Additionally, the 3D EIoU loss considers the three sides' changes with the midpoint distance between the two 3D bounding boxes, the resulted performance is better than CIoU. Additionally, 3D EIoU may be used as criteria in non-maximum suppression (NMS), which considers both the distance between the midpoint of two bounding boxes and the three geometric sides' changes while suppressing redundant boxes. To evaluate our proposed methods, EIoU loss and EIoU-NMS are incorporated into the PointPillars network for

measuring object detection performance on the KITTI dataset[11][5].

In specific, the key influences of this study can be briefly outlined as follows. Firstly, a Complete-IoU loss, i.e., 3D CIoU loss, is proposed for bounding box regression, which has faster convergence than 3D IoU, 3D GIoU and 3D DIoU losses. Secondly, a 3D Distance-IoU loss, i.e., 3D DIoU loss, is proposed for 3D bounding box regression which has faster convergence than 3D IoU, and GIoU losses. Thirdly, a 3D Efficient-IoU loss, i.e., 3D EIoU loss, is proposed which takes the three sides' changes with central distance between two boxes midpoint, the 3D EIoU has faster convergence than other loss functions. Fourthly, the 3D EIoU function is added to the PointPillars network for optimizing bounding box regression. Finally, the EIoU-NMS process is performed to remove the duplicated bounding box. The detection pipeline for the network is tested on the KITTI benchmark dataset and caused the network to be better than others for average precision (AP)values.

The paper is structured as follows. The related work is reviewed in Section II. Driving the 3D loss function CIoU, DIoU, and EIoU is discussed in Section III. The experiment and exploration for object detection, by using 3D EIoU illustrated in Section IV, followed by conclusion.

II. RELATED WORK

According to how the input data is represented, there are three different types of 3D object detection techniques: monocular based on image, second based on point cloud, and third based on fusion inputs.

A. Monocular image-based detection techniques

These methods have lack of 3D space information, which is the most challenging issue. However, studies [1][2] focused on these methods because the used tools for obtaining the monocular images are suitable and cheap. The [1] study obtained 3D proposal objects from the predefined 3D regions, where the objects must be on the ground area to take 3D proposal objects from monocular images. Contextual information, for each candidate, the typical object's size, shape, and prior location are scored to select the best candidates. Research [2] solved the calculating charge of 3D sliding windows in [1] where the category, position, and the orientation angle of the 2D bounding box are predicted firstly in a monocular-based image. At that time the size value of the 3D box and position coordinate points are estimated in the camera coordinates plane. The generated 3D bounded box is projected in three views, Front, side view, and Bird's Eye View map (BEV). The 2D box's texture information was combined with the 3D structural features that were extracted from projected surface regions. The performance of the detection process improved, and the 3D bounding box refined, by using the fused features. This process obtains superior performance but the detecting precision was far away from achieving the necessities of automatic driving and many other usages.

B. Point Cloud-based Detection Techniques

The real world scenes directly reflected by point cloud data. In contrast to monocular pictures, point cloud data include critical information for 3D object detection. Due to the irregularity of the 3D point cloud data, deep learning cannot be utilized to immediately detect an object in the point cloud. There are two methods that are frequently used to transform point clouds into consistent data, after which the data is put to a 3D object detection network. The first method achieves 2D images [6][7][12]by projecting the point cloud data to the 2D plane. Complex Yolo [6] and PIXOR [7] projected point cloud data to bird's eye view and applied a 2D detecting process on the projected image, the point cloud here was utilized efficiently. However, the detection performance was poor where the point cloud's spatial structural evidence was lost.

Some studies like [13] [14] [15], convert point cloud data to 3D voxel grids, without projecting the point cloud to 2D plane's coordinates. VoxelNet [13] and SECOND [14] are one-stage detectors. After the voxels process, the whole point cloud denoted by a four-dimension tensor. The region proposal network (RPN) is the final 3D convolutional layer that handles this tensor in a sequential manner [14], the RPN calculated classification value and bounding box regression map. Voxelization used in this study to convert point clouds into regular data for the 3D object detection process.

C. Image-point cloud fusion-based detection techniques

In various studies, RGB picture and depth map have been merged for 3D object detection [16][17][18][19]. In the [17] study, the convolutional neural networks took color features from RGB images, and from the depth map, it extracted geometric features. Then advanced visual features and geometric features were extracted using deep belief networks (DBN). These learned features are used to obtain a fused feature for object detection.

In the [18] study, the CNN was used to extract the geometric and appearance features from the depth and RGB images correspondingly, obtaining 2D detecting results from RGB images. The resulted 2D bounded boxes joined with geometric features, classification outcomes with the final results transformed to 3D space. Additionally, the bounding box regression was useful for refining the 3D boxes. However, the [19] study directly combined geometric and appearance feature, which were then used to determine the final detection results. These techniques slowed down the detecting process and increased the cost of calculation. In these methods, the appearance and geometry features were extracted in various ways.

Some methods [3][4][20][21] fused point cloud with RGB image for 3D object detection. Typically, [3] work represented a point cloud as a bird's eye view (BEV). More information was collected when the detecting system was fed the FV and BEV of the point cloud with the RGB picture. The BEV suffers less occlusion, which was useful for the 3D object detection process. The resulted object is projected to FV and RGB images. The learned features from these views fused for bounding box regression and object classification. [4] is a two-stage 3D object detector that used RGB photos to identify objects; in the first stage, 2D detecting boxes created, and in the second stage, the 2D boxes projected into a point cloud to create a point cloud frustum. Finally, the results were segmented, and 3D bounding boxes were calculated.

According to the research mentioned above, the projected 3D bounding box strongly relies on the 2D area proposal

network to perform better. Unlike these approaches, to increase the precision of 3D object detection, we exclusively used point cloud data.

III. METHOD

In this section, we propose the 3D Complete IoU, Distance IoU, and Efficient IoU loss functions. By taking the three coordinate points x,y,z and the three boxes' sides width, height and length, while the previous works [9][10] find these loss functions for 2D boxes with x,y coordinate and using only two sides for boxes, width and height only.

A. Distance-IoU loss function

For minimizing the standardized distance between center points of two 3D bounded boxes distance-iou loss function used, which has the following penalty term

$$\mathcal{R}_{DIoU} = \frac{p^2(b,b^{gt})}{c^2}, \quad (1)$$

where B and B^{gt} 3D boxes have central points denoted by b and b^{gt} in the above equation. The $p(.)$ denotes the Euclidean distance, and the c value denotes the diagonal length of the smallest enclosing 3D box that contains the two boxes. Then the DIOU loss function defined as

$$L_{DIOU} = 1 - IOU + \frac{P^2(b,b^{gt})}{C^2}. \quad (2)$$

The DIoU loss straightly decreases the distance between central points. The DIoU loss function is invariant to the scale of the bounding box.

B. Complete IoU loss function

The CIOU loss function takes into account three geometrical components. These factors are the overlap area, the midpoint distance, and the aspect ratio [9]. By defining the 3D predicted box B and the 3D targeted box B^{gt}, the CIOU loss equation is written as follows.

$$L_{CIOU} = 1 - IOU + \frac{P^2(b,b^{gt})}{C^2} + \alpha v. \quad (3)$$

Both B and B^{gt}3D boxes have central points denoted by b and b^{gt} in the above equation. The Euclidean distance P is measured and is calculated as follows.

$$P(.) = ||b - b^{gt}||_2 \quad (4)$$

The C value denotes the diagonal line value of the smallest enclosed 3D box covering the two boxes. The α is a positive trade-off factor, and v processes the uniformity of aspect ratio,

$$v = \frac{4}{\pi}(\tan^{-1}\frac{h}{\sqrt{l^2+w^2}} - \tan^{-1}\frac{h_{gt}}{\sqrt{l_{gt}^2+w_{gt}^2}})^2. \quad (5)$$

and $\alpha = \frac{v}{(1-IOU)+v}$, measures the discrepancy of the height to length and width ratio. For the final optimization of CIoU loss, the partial derivative of v to height (h), width (w), and length (l) is calculated as follows.

$$\frac{\partial v}{\partial h} = (\tan^{-1}\frac{h}{\sqrt{l^2+w^2}} - \tan^{-1}\frac{h_{gt}}{\sqrt{l_{gt}^2+w_{gt}^2}}) \frac{8}{\pi^2}[\frac{\sqrt{l^2+w^2}}{l^2+w^2+h^2}], \quad (6)$$

$$\frac{\partial v}{\partial l} = \left(\tan^{-1}\frac{h}{\sqrt{l^2+w^2}} - \tan^{-1}\frac{h_{gt}}{\sqrt{l_{gt}^2+w_{gt}^2}}\right)\frac{-8}{\pi^2}[\frac{hl(l^2+w^2)^{-\frac{1}{2}}}{l^2+w^2+h^2}], \quad (7)$$

$$\frac{\partial v}{\partial w} = (\tan^{-1}\frac{h}{\sqrt{l^2+w^2}} - \tan^{-1}\frac{h_{gt}}{\sqrt{l_{gt}^2+w_{gt}^2}})\frac{-8}{\pi^2}[\frac{hw(l^2+w^2)^{-\frac{1}{2}}}{l^2+w^2+h^2}]. \quad (8)$$

Both converge speed and detection accuracy improved by using the CIOU loss compared to previous loss functions. However, the term v in L_{CIOU} it still slows down the convergence speed of two bounding boxes.

C. 3D EIoU loss function

We proposed an efficient version of 3D IoU loss, i.e., a 3D EIoU loss. This 3D EIoU loss function can handle three axis-aligned bounding boxes, where the EIoU loss is defined as follows,

$$L_{EIoU} = L_{IoU} + L_{dis} + L_{asp}, \quad (9)$$

$$= 1 - IoU + \frac{p^2(b,b^{gt})}{c^2} + \frac{p^2(w^p,w^{gt})}{C_w^2} + \frac{p^2(h^p,h^{gt})}{C_h^2} + \frac{p^2(l^p,l^{gt})}{C_l^2}. \quad (10)$$

where C_w, C_h, and C_l are the width, height, and length of the smallest box that encloses the two boxes. The above loss equation is divided into three parts: the IoU loss $LIoU$, the distance loss $Ldis$, and the aspect loss $Lasp$. The CIoU loss and DIoU loss characteristics can be observed in the above equation. The EIoU loss directly reduces the difference in width, height, and length between the target and anchor boxes, which leads to faster convergence speed and better localization. It is essential to examine the relationship among loss functions by conducting the simulation experiment.

D. Non-Maximum Suppression with EIoU

In the proper NMS process, the IoU value was utilized to remove the redundant detected boxes. However, the overlap area may yield incorrect suppression for the occlusion situation. In this study, we propose that EIoU is a more suitable technique for NMS, by using EIoU, the overlap area, the three sides' changes, and the midpoint distance value between the two bounding boxes must be measured in the suppression process. The 3D bounded box with the highest confidence score is preserved, and any nearby bounded boxes are eliminated.

IV. EXPERIMENTATION PROCESS AND RESULTS

All detection trained, and all results on standard object detection benchmarks are reported, on the KITTI dataset. The proposed 3D EIoU algorithm evaluates by applying it to PointPillars, one-stage object detection instructions.

A. The Experimental Setting

The experimental area in this work organized as follows: laptop with Asus Ryzen 9 CPU (4.6 GHz, 8 cores), 40 GB RAM, Ubuntu 20.04 64-bit operating system. NVIDIA GeForce RTX 3070 laptop GPU 8GB, Cuda V11.1.

B. Dataset and Training Step

Two subsets provided by the KITTI dataset, one of them contain 7,481 point cloud files and image files, used for training, the other subset contains 7,518 used for testing. Due to the inaccessibility of the original test subset's ground truth, we divide the original training subset into new training and validation sets. As a consequence, we are able to obtain 3,712 samples of data for training and 3,769 samples of data for validation. Three difficulty levels hard, moderate, and easy,

have been assigned to the object detection on the KITTI benchmark. For each valid frame, the point cloud has been utilized to detect objects.

We train the original PointPillars ideal model using the authors' code that was made available with the same config file (xyres16.config) to obtain the baseline outcomes for using smooth l1 loss as a regression loss function. We compare these baseline results with the results generated from using the EIoU loss function. And for the second time, we compare the baseline results by adding the NMS-EIoU process to the Pointpillars model.

C. Evaluation Step and Aanalysis Detection Outocomes

In this experiment, the $AP|_{R40}$ metric used for evaluation, we calculate the $AP|_{R40}$ by a certain IoU threshold for each difficulty class to get the localization and precision accuracy for each object. IoU threshold for a car is equal to 0.7 and for cyclists and pedestrian is equal to 0.5. In Table I we measure the relative improvement between the two biggest values. From Table I, EIoU loss combined with EIoU-NMS brings a marvelous improvement of 1.53% $AP|_{R40}$, 1.7369% $AP|_{R40}$, and 1.781% $AP|_{R40}$ for class cyclist. While the $AP|_{R40}$ values for pedestrian class improved by 1.6084%, 1.3869, and 1.633% for each difficulty easy, moderate, and hard correspondingly. The class car did not exhibit any improvement after the addition of the combined EIoU loss and EIoU-NMS, indicating that EIoU loss and EIoU-NMS are only appropriate for tiny bounding boxes with 3D geometric dimensions. The cyclist and pedestrian classes have small 3D bounding boxes compared with car class. We can see this very clearly in the results in Fig.1, Fig.2, and Fig.3. The outcomes of the bird's eye view map (BEV) are displayed in Table II. By adding the EIoU loss with NMS-EIoU to PointPillars, the class cyclist improved its $AP|_{R40}$ values, 0.6861% for easy, 1.1402% for moderate and 0.7205% for hard. While the class pedestrian gets the following $AP|_{R40}$ improved values, for easy, it is equal to 1.2782%, moderate improved by 1.213% and hard difficulty improved by 1.2936%. The class car did not get any improvement because of its big bounding boxes compared to cyclist and pedestrian classes.

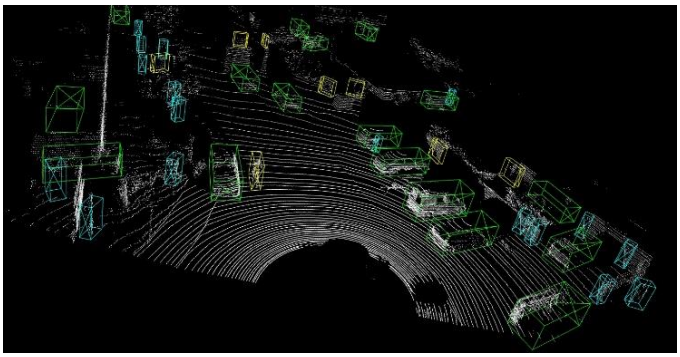

Fig.1 Car class with a green label

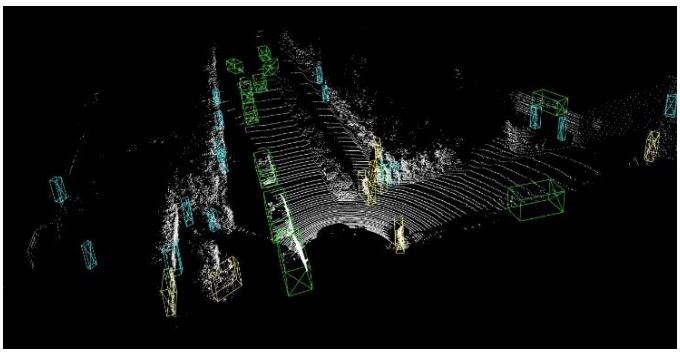

Fig.2 Cyclist class with a yellow label

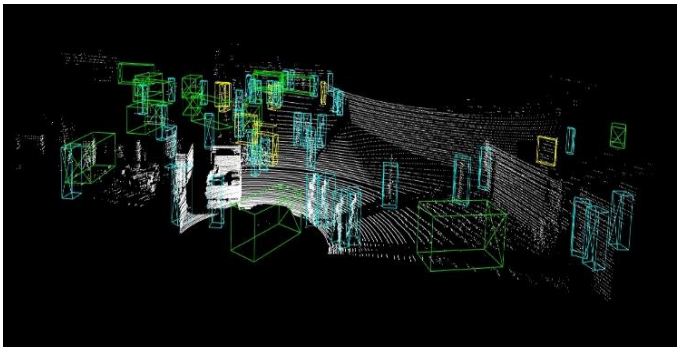

Fig.3 Pedestrian class with a blue label

Table I. The evaluation outcomes by training PointPillar with smooth-l1, EIoU and NMS-EIoU losses. The outcomes are calculated on the KITTI val set on the three difficulties for each class which have 3D boxes

	Car			**Cyclist**			**Pedestrian**		
Method	***Easy***	***Mod.***	***Hard***	***Easy***	***Mod.***	***Hard***	***Easy***	***Mod.***	***Hard***
PointPillars	87.4355	78.3913	75.4573	80.7919	63.8438	59.6073	54.9425	49.1772	44.7744
PointPillars+EIoU	87.8606	78.3795	75.7117	82.4072	63.7676	59.6324	54.3985	48.7929	44.5937
PointPillars+EIoU+NMS-EIoU	87.4358	78.1	75.2073	83.9372	65.5807	61.4134	56.5509	50.5641	46.2267
Relative improvement %	-0.4248	-0.0118	-0.2544	1.53	1.7369	1.781	1.6084	1.3869	1.633

Table II. The evaluation outcomes by training PointPillar with smooth-l1, EIoU and NMS-EIoU losses. The outcomes are calculated on the KITTI val set on the three difficulties for each class which have BEV boxes

	Car			**Cyclist**			**Pedestrian**		
Method	***Easy***	***Mod.***	***Hard***	***Easy***	***Mod.***	***Hard***	***Easy***	***Mod.***	***Hard***
PointPillars	91.6571	87.9629	86.9057	87.3822	68.978	64.7245	60.9621	54.7965	50.7874
PointPillars+EIoU	91.9859	88.0358	86.9768	87.8581	68.8679	64.2836	59.8845	54.1707	49.9252
PointPillars+EIoU+NMS-EIoU	91.614	87.7334	86.4052	88.5442	70.1182	65.445	62.2403	56.0095	52.081
Relative improvement %	-0.3288	-0.0729	-0.0711	0.6861	1.1402	0.7205	1.2782	1.213	1.2936

D. Relationship among Loss Function

For finding the best loss function among the explained functions, we conduct a simulation experiment. The relations between bounded boxes are mostly in terms of aspect ratio, distance, and scale are covered in the simulation experiment. In specific, seven 3D boxes with several aspect ratios are utilized (i.e.,1:1:1, 0.33:1:1, 1:0.33:1, 1:1:0.33, 1.5:1:1, 1:1.5:1 and 1:1:1.5) as 3D target boxes. The seven 3D target boxes' center points are set at (5,5,5), see Fig.4. In a circle with a radius of 3 and centered at (5,5,5), the 3D anchor boxes are equally distributed at 1,000 points. The 1,000 points are correspondingly chosen to place the 3D anchor boxes with seven aspect ratios and seven scales. In these situations, non-overlapped and overlapped 3D boxes are involved. At each point, the volumes of the 3D anchor boxes are set to 0.5, 0.67, 0.75, 1, 1.33, 1.5, and 2. For a given point and scale, seven aspect ratios are considered, i.e., following the same setting with 3D target boxes (i.e., 1:1:1, 0.33:1:1, 1:0.33:1, 1:1:0.33, 1.5:1:1, 1:1.5:1 and 1:1:1.5). All the 1,000 x 7 x 7 3D anchor boxes should be fitted to each 3D target box. Totally the regression cases will be equal to 343,000 = 7 x 7 x 7 x 1,000, see Fig.4.

Algorithm I:Simulation Experiment

Input: Loss $\mathcal{L}$ function with altered target attributes constraints and iteration T.

$M = \{\{B_{n,s}\}_{s=1}^{S}\}_{n=1}^{N}$ is the set of anchor boxes at N=1,000 equally spread points within the circular area with center (5,5,5) and radius 3 and S= 7 x7 covers seven scales and seven aspect ratios of anchor 3D boxes.

$M^{gt} = \{B_i^{gt}\}_{i=1}^{7}$ is the set of target 3D boxes that are positioned at (5,5,5), and have seven aspect ratios.

Outcome: Regression error $E \in \mathbb{R}^{TxN}$

1: Initialize $E = 0$ and maximum iteration value T.
2: Do bounded box regression:
3: for $n = 1$ to N do
4: for $s = 1$ to S do
5: for $i = 1$ to 7 do
6: for $t = 1$ to T do

$$7\ \eta = \begin{cases} 0.5 & if\ t <= 0.8\,T \\ 0.05 & if\ 0.8T < t <= 0.9\,T \\ 0.005 & if\ t > 0.9\,T \end{cases}$$

8: find G_n^{t-1} which is the gradient of $\mathcal{L}(b_n^{gt}, b_n^{t-1})$
9: $b_n^t = b_n^{t-1} + \eta\, G_n^{t-1}$
10: $e_n^t = e_{n-1}^t + \left|b_n^t - b_n^{gt}\right|$
11: for End
12: for End
13: for End
14: for End
15: return E

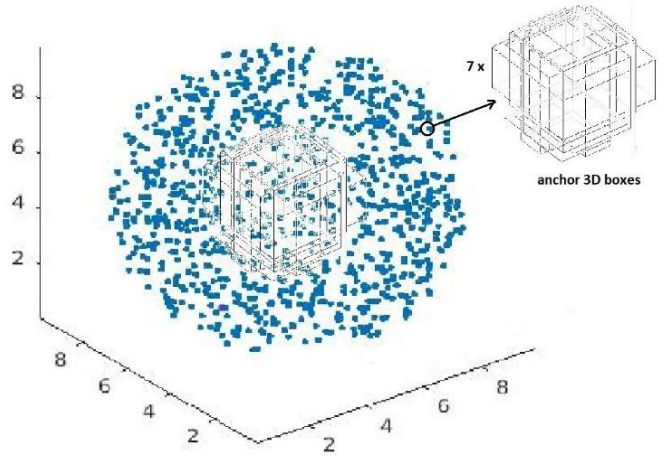

Fig.4 The 343,000 regression cases adopt by examining several distances, aspect ratios, and scales.

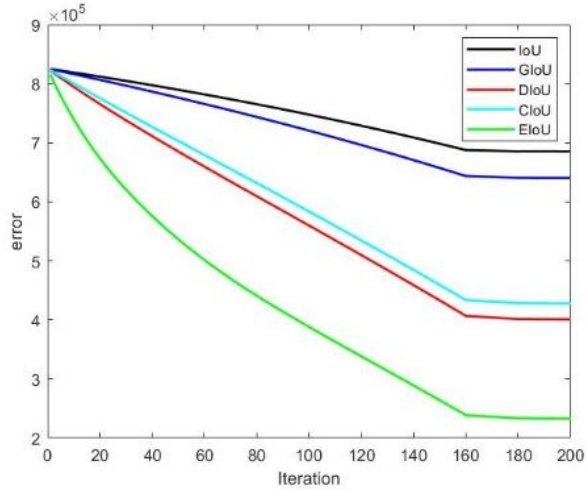

Fig.5 Multiple loss equations' regression error graph lines at iteration t.

The performance illustrated in Fig.5 is the result of running the above-mentioned simulation with an iteration number of 200. The error value decreases for the EIoU loss function, indicating proper behavior for this loss function.

V. CONCLUSIONS

In this study, we presented three 3D losses—CIoU loss, DIoU,and EIoU loss—for bounding box regression as well as EIoU-NMS for reducing redundant detection boxes. The 3D EIoU loss function can reach quicker convergence than the 3D DIoU, GIoU, and IoU loss functions by directly decreasing the distance between the midpoint of two 3D bounding boxes. Additionally, the 3D EIoU loss takes three geometric sides' changes into account, and leads to faster convergence and better performance than the 3D CIoU loss function. The 3D EIoU-NMS can be simply integrated to PointPillars object detection pipeline, and reach improved outcomes on dataset. We believe that the offered 3D EIoU-NMS loss function will be of great value to bounding boxes with small dimensions. In the future, we can test the new loss function on other object detection networks to evaluate its performance.

ACKNOWLEDGMENT

The research work is supported by research grants: FRGS/1/2020/STG07/UKM/02/3 and TAP-K021588.

REFERENCES

[1] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3D Object Detection for Autonomous Driving," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2147–2156. doi: 10.1109/CVPR.2016.236.

[2] B. Li, W. Ouyang, L. Sheng, X. Zeng, and X. Wang, "GS3D: An Efficient 3D Object Detection Framework for Autonomous Driving," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1019–1028. doi: 10.1109/CVPR.2019.00111.

[3] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017, vol. 2017-Janua, pp. 6526–6534. doi: 10.1109/CVPR.2017.691.

[4] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D Object Detection from RGB-D Data," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 918–927. doi: 10.1109/CVPR.2018.00102.

[5] "Kitti 3D Object Detection Benchmark Leader Board." http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d.

[6] M. Simony, S. Milzy, K. Amendey, and H.-M. Gross, "Complex-yolo: An euler-region-proposal for real-time 3d object detection on point clouds," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, p. 0.

[7] B. Yang, W. Luo, and R. Urtasun, "PIXOR: Real-time 3D Object Detection from Point Clouds," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7652–7660. doi: 10.1109/CVPR.2018.00798.

[8] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 658–666. doi: 10.1109/CVPR.2019.00075.

[9] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," in *Proceedings of the AAAI conference on artificial intelligence*, 2020, vol. 34, no. 07, pp. 12993–13000.

[10] Y.-F. Zhang, W. Ren, Z. Zhang, Z. Jia, L. Wang, and T. Tan, "Focal and efficient IOU loss for accurate bounding box regression," *arXiv Prepr. arXiv2101.08158*, 2021.

[11] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361. doi: 10.1109/CVPR.2012.6248074.

[12] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3d lidar using fully convolutional network," *arXiv Prepr. arXiv1608.07916*, 2016.

[13] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4490–4499. doi: 10.1109/CVPR.2018.00472.

[14] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[15] M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong, and I. Posner, "Vote3Deep: Fast object detection in 3D point clouds using efficient convolutional neural networks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1355–1361. doi: 10.1109/ICRA.2017.7989161.

[16] K. Shin, Y. P. Kwon, and M. Tomizuka, "RoarNet: A Robust 3D object detection based on region approximation refinement," in *IEEE Intelligent Vehicles Symposium, Proceedings*, 2019, vol. 2019-June, pp. 2510–2515. doi: 10.1109/IVS.2019.8813895.

[17] W. Liu, R. Ji, and S. Li, "Towards 3D object detection with bimodal deep Boltzmann machines over RGBD imagery," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3013–3021. doi: 10.1109/CVPR.2015.7298920.

[18] Z. Deng and L. J. Latecki, "Amodal Detection of 3D Objects: Inferring 3D Bounding Boxes from 2D Ones in RGB-Depth Images," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 398–406. doi: 10.1109/CVPR.2017.50.

[19] Q. Luo, H. Ma, L. Tang, Y. Wang, and R. Xiong, "3d-ssd: Learning hierarchical features from rgb-d images for amodal 3d object detection," *Neurocomputing*, vol. 378, pp. 364–374, 2020.

[20] R. Huitl, G. Schroth, S. Hilsenbeck, F. Schweiger, and E. Steinbach, "TUMindoor: An extensive image and point cloud dataset for visual indoor localization and mapping," in *2012 19th IEEE International Conference on Image Processing*, 2012, pp. 1773–1776. doi: 10.1109/ICIP.2012.6467224.

[21] M. Li, Y. Hu, N. Zhao, and Q. Qian, "One-stage multi-sensor data fusion convolutional neural network for 3d object detection," *Sensors*, vol. 19, no. 6, p. 1434, 2019.