Journal of Physics: Conference Series

PAPER • **OPEN ACCESS**

A Survey on Monocular 3D Object Detection Algorithms Based on Deep Learning

To cite this article: Junhui Wu *et al* 2020 *J. Phys.: Conf. Ser.* **1518** 012049

A Survey on Monocular 3D Object Detection Algorithms Based on Deep Learning

Junhui Wu[1, a], Dong Yin[1, b], Jie Chen[1, c], Yusheng WU[2, *], Huiping Si[1, d] and Kaiyan Lin[1, e]

[1]Institute of Modern Agricultural Science & Engineering, Tongji University, Shanghai, China
[2]Equipment Management Department, Xiamen Tobacco Industrial Co., Ltd. Fujian, China

Email: [a]junhui_wu@163.com; [b]tjyd@tongji.edu.cn; [c]chenjie18@yahoo.com.cn; [*]21480276@qq.com; [d]sihuiping@tongji.edu.cn; [e]ky.lin@163.com

Abstract. An accurate and effective perception of environment is important for autonomous vehicle and robot. The perception system needs to obtain the 3D information of objects, which includes objects' space location and pose. Camera is widely equipped on autonomous vehicle because of its price advantage. However, the monocular camera cannot provide depth information which is necessary for 3D object detection. Many algorithms based on monocular 3D object detection have been developed in recent years. Deep learning is popular for perception system which transforms image data from camera into semantic information. This paper presents an overview of monocular 3D object detection algorithms based on deep Learning and summarize the contributions and limitations of these algorithms. We also compare the performance of different algorithms on different datasets.

1. Introduction

3D Object detection is an important research problem in deep learning. In many fields, such as autonomous driving and robotic grasping, system must obtain the 3D information of objects through sensors so that it can work normally. The current sensors of 3D object detection can be divided into vision, lidar, and multi-sensors. Algorithms based on lidar are accurate and effective, but the high cost limits its use in industry. Algorithms based on vision can also be divided into two types: monocular and binocular. Algorithms based on vision are widely used because of its low cost and rich texture information. The biggest disadvantage of monocular vision is that it cannot directly obtain the depth information of the image, which may result a deviation in 3D location and 3D size estimation in the monocular object detection. The cost of binocular vision is more than monocular vision. Binocular vision can provide more accurate depth information than monocular vision, but the range of visual field is narrow, which cannot be satisfied under some working conditions.

This paper presents an overview of monocular 3D object detection algorithms and is structured as follows. Section 2 introduces algorithms based on deep learning. We divide these algorithms into six types according to their different characteristics. Section 3 compares the performance between these algorithms on different datasets.

2. 3D OBJECT DETECTION ALGORITHMS

Monocular 3D object detection is actually an ill-posed problem. The monocular image lacks depth information because of the principle of perspective transformation. In order to achieve monocular 3D detection well, many algorithms have been developed in recent years.

Some algorithms detect specific kinds of objects, which use some prior hypotheses and template matching. And some algorithms use deep learning to predict the depth map of the image first, which serves as a basis for 3D object detection in the next stage. The PnP based algorithms, which establish the correspondence between 3D key points on the 3D model and the 2D key points on the monocular image, can achieve good detection results. The latest algorithm is to convert the image data format into point clouds data format, and then use the deep learning networks for processing point clouds to predict the 3D information of the objects. In this section, we will present some algorithms of monocular 3D object detection. we summarize them into six types according to their characteristics and introduce the contributions and limitations of these algorithms in Table 1.

2.1. Prior information fusion based Algorithm

The algorithm based on the prior information fusion is to establish the correspondence between 2D and 3D. Prior knowledge include semantic information, context information, shape information and location information, etc. These prior information are only applicable to specific objects.

Chen[1] first proposed the Mono3D algorithm in 2016, which mainly uses two stages. Firstly, dense sampling is performed according to the prior hypothesis, and a series of 3D object candidate 3D boxes are generated. Then the generated 3D candidate boxes are re-projected to generate the object's 2D detection boxes. Mono3D use the 2D object detection network, Faster-RCNN to extract the corresponding features. The semantic, contextual information, the shape information of the detected object, and the prior information of the position are used to calculate the energy function of the detection boxes. The loss function can finally optimize an accurate 3D box.

Mono3D[1] first used a priori information to extract 3D boxes. However, there will be a problem of error accumulation in the process of calculating the energy loss function. Therefore, as a pioneer algorithm of 3D object detection, this algorithm cannot achieve satisfactory detection accuracy and detection speed. Mono3D++[2] is based on Mono3D, and uses the ceres algorithm for optimizing the 3D object detection results.

2.2. Template matching based Algorithm

The template matching based algorithm are to extract 3D information through CAD template matching, which is difficult to process multi-object situation. During this detecting process, a template library will be established, and the network will match the best model in the template library for the objects in the image.

Chabot[3] proposed the Deep-MANTA algorithm, which uses a multi-tasking network structure. This algorithm predefines a set of key points for the vehicle, including the lights, rear view mirrors, etc. The convolutional neural network of regression returns the 2D bounding box of the objects and the key points which are predefined on the vehicle. Finally, the artificial 3D vehicle model library is used for matching, and then the networks can obtain the 3D information of the objects. In the latest work, the ROI-10D algorithm[4] is based on the 2D detection network structure of Resnet-FPN, which combines deep feature maps to obtain shape dimensions using CAD models. At last the network performs the shape regression of the object in the shape dimension, and can get the specific object 3D information.

2.3. Depth estimation based Algorithm

On the basis of existing depth estimation networks, many 3D object algorithm treat these depth estimation algorithms as a sub-module of their own networks. Depth estimation can make up for the shortcomings of monocular vision and detect the 3D information of objects more accurately.

The popular depth estimation network is DORN[5], which combines multi-scale features to predict pixel-by-pixel depth with low errors. Another popular depth estimation network is unsupervised depth estimation algorithm[6], which uses the left and right image for reconstruction matching during the training stage, and only the left images are used during inference stage. Based on the monocular depth estimation, Xu[7] proposes the MF3D algorithm, which combines the depth estimation algorithm and Deep3Dbox[8]. The object region of interest and depth feature map are fused to calculate the object coordinate and Spatial location information. MonoGRNet[9] also uses depth estimation, but it differs from MF3D because MonoGRNet[9] uses an instance-level depth estimation algorithm. This network only performs depth estimation on the image area where the objects exists. This method avoids the depth estimation in the entire picture, which can reduce the amount of calculation.

Other algorithms use the geometric constraints of rigid objects such as vehicles to calculate the spatial position of the object. OFT-NET[10] uses the correspondence between the image space and the 3D space, establishes an orthogonal transformation between image space and 3D space. Then the network converts the features of the image to the bird's-eye view through Inverse perspective transform. Finally, the residual network unit is used to process the feature map of the bird's-eye view to obtain the final result.

2.4. Single-stage 2D detection based Algorithm

This algorithms are currently popular, because this algorithms can use the good results of the current 2D detection of deep networks and add some effective improvements on this basis to achieve good 3D results. This algorithm can usually achieve good results. However, it is difficult to make a breakthrough in previous work.

Mousavian[8] proposes a 3D object detection algorithm of Deep3Dbox. This algorithm extends the existing 2D detection network, and uses the regression algorithm to directly return the object's spatial size and its yaw angle. A major contribution of Deep3Dbox[8] is to propose the Multibins skill, which calculates the yaw angle of object.

The previous algorithm mainly uses the L2 loss function to directly return to the yaw angle, while Multibins first discrete the yaw angle into multiple overlapping 3D bins, and then using a convolutional neural network to predict the confidence of each bin and the offset from the rotation residual of the base bin. In the estimation of the object space size, the L2 loss function is directly used to calculate the offset of the space size. After the 3D size and yaw angle of the object are determined, the 6d pose of the objects can be restored by calculating the rotation matrix of the object in the camera coordinate system.

PoseNet[11] uses a convolutional neural network to directly return the pose of the camera from the monocular image. However, it is difficult to directly return to the spatial position of the object, because of the lack of depth information and the need for a large search space. Therefore, PoseCNN [12] detects the position of an object in a 2D image while also predicting its depth so that it can obtain its 3D position. XIANG[12] extended the 2D object detection network SSD to directly predict the 6D pose of object.

Buyu[13] proposes GS3D, which adds a feature extraction module for visible surfaces based on Deep3Dbox[1]. An important idea of this algorithm is to first propose a rough 3D box estimation, and then use the relevant Features to optimize. Similar to this algorithm is the work of FQNet[14], which first uses a 2D boxes to select a suitable rough 3D boxes in space, and then samples many 3D based on this, and then uses IOUNet to obtain the best 3D boxes.

PoseCNN[12] is also difficult to estimate 3D rotation directly, because the rotation space is non-linear, making it difficult for CNN to recognize space rotation. In order to avoid this problem, some algorithms have been proposed to discretize the rotation space. This transformation can convert the direct regression 3D rotation into a classification task, but at the same time, this discretization will also cause the result to be inaccurate. Post-processing is very important. M3D-RPN[15] uses the shared 2D and 3D detection space to build an independent monocular 3D area recommendation network, which can achieve the best performance at the current stage.

2.5. PnP based Algorithm

The algorithm of using key points is not to directly obtain the pose of the object from the monocular image, but use a two-stage algorithm. The network first predicts the 2D key points of the object, and then calculates the pose of the object by 2D-3D correspondence with the PnP algorithm[16].

2D keypoint detection is relatively easier than 3D localization and rotation estimation, but requires a model of a known 3D object and some predefined keypoints. For objects with rich textures, traditional algorithms can detect local key points more robustly, even in cluttered scenes and severe occlusions. However, the traditional algorithm will have some difficulties for images without textures and low resolution. TEKIN[17] uses the YOLO architecture to estimate the key points of the object. The network is based on feature map with low resolution. But when the object is occluded, it has a bad impact on the result.

In order to solve the problem of occlusion and truncation, the HU[18] introduced a segmentation algorithm to estimate the 6D pose of an object, combining multiple local pose estimates, which can also produce better keypoint predictions in the presence of large occlusions. PVNet[19] uses the dense algorithm in the prediction stage of key points. Each pixel votes for the predefined key points, which are optimized by ceres algorithm. PVnet can get good results comparing the previous algorithm.

Table 1. Contributions and Limitations of six types of monocular vision algorithms

Algorithm Category	**Algorithm name**	**Contributions**	**Limitations**
Prior information fusion	Mono3D Mono3D++	Integrating a large amount of prior information	unsatisfactory detection accuracy and detection speed
Template matching	Deep MANTA ROI-10D	Extracting 3D information through CAD template matching	complex procedure and difficult to process multi-object situation
Depth estimation	MF3D MonoGRNet OFT-NET	Fusion 2D detection network and depth estimation network	inaccurate depth estimation resulting error accumulation
2D-3D single-stage	Deep3Dbox PoseNet PoseCNN M3D-RPN	Combining with 2D detection network, adding 3D regression branches	Non-linear rotation space regression resulting inaccurate results
PnP	TEKIN Seg-driven PVNet	First detecting the predetermined key points, and then using the PnP algorithm to calculate the attitude	Key point detection affecting the detection results and requiring post-processing
Pseudo point cloud	Pseudo-LiDAR AM3D MonoPSR	Transforming image data into point cloud data	requiring high-resolution images and unsatisfactory detection speed

2.6. Pseudo point cloud based Algorithm

The above algorithms are based on image information and establish the relationship between 2D and 3D. Another popular algorithm is to convert the image information into point cloud information, and then use the point cloud-related network for processing. This algorithms propose that the point cloud data format is more suitable for 3D object detection than image. So it can achieve satisfactory detection results using only the camera.

The PointNet network[20] can be used for point cloud classification and semantic segmentation. The way of extracting features in PointNet is global, which is different from the way of convolutional neural network to extract local features layer by layer. Based on this idea, Charles[21] proposes PointNet++, which can extract feature layers at different scales in local features. PointNet [20] and PointNet++ [21] are mainly for the classification and segmentation of point clouds.

Based on these two networks, QICR[22] proposes Frustum-PointNet, which is used to 3D objects detection. Pseudo-LiDAR[23] proposes that the accuracy limit of monocular image detection of 3D objects is not because the accuracy of monocular depth estimation is not sufficient, but because the data representation of point clouds is more suitable for 3D object detection than images. Therefore, Pseudo-LiDAR[23] uses the DORN network to estimate the depth, and uses the corresponding mathematical relationship to convert the image information into pseudo-point cloud information. And then it uses two more advanced point cloud processing networks to process the data of the pseudo-point cloud.

The MF3D introduces a multi-layer fusion scheme to generate the final pseudo-point cloud information. And the MonoPSR[24] introduces an instance reconstruction module that first predicts the local point cloud of the object. The point cloud information is then projected back into the world coordinate system using the rotation matrix. In this module, three effective loss functions are constructed to realize the recognition of 3D objects. The AM3D[25] is the algorithm that uses the monocular image to get the best results at this stage, which proposes a technique to fuses the image information and the pseudo point cloud information and can get satisfactory results.

3. Evaluation

This section presents performance for some of the reviewed algorithms. Different datasets have different metrics for performance of algorithms in different scenarios. The KITTI dataset[26] and Apollocar3D dataset[27] mainly contain data of outdoor objects, such as vehicles and pedestrians. For 3D object detection in KITTI dataset, $\boldsymbol{AP_{3D}}$ and $\boldsymbol{AP_{BEV}}$ are the most commonly used. $\boldsymbol{AP_{3D}}$ can evaluate 3D object detection results according to different 3D IOU (intersection of union) thresholds. The commonly used 3D IOU thresholds are 0.25, 0.5 and 0.7. $\boldsymbol{AP_{BEV}}$ transforms the 3D detection results into a bird's-eye view, which evaluates the accuracy of location.

We show the performance of the optimal algorithm for each type of algorithm on KITTI in Table 2. According to Table 2, the accuracy of prior information fusion and template matching algorithm is not as good as the accuracy of other algorithms. These two algorithms are relatively early proposed whose process is complicated, and the amount of calculation is much. The accuracy of pseudo point cloud is currently the best, although this algorithm is temporarily unable to achieve single-stage.

Table 2. Performance of representative algorithms on KITTI dataset

Algorithm name	**Category**	$\boldsymbol{AP_{3D}}$**(IOU=0.7)** **Easy / Moderate /Hard**	$\boldsymbol{AP_{BEV}}$**(IOU=0.7)** **Easy / Moderate /Hard**
Mono3D++	prior information fusion	10.6%/7.9%/5.7%	16.7%/11.5%/10.1%
ROI-10D	template matching	12.30%/10.30% /9.39%	16.77%/12.40%/11.39%
MonoGRNet	depth estimation	13.88%/10.19%/7.62%	/
M3D-RPN	single-stage 2D detection	20.27%/17.06%/15.21%	25.94%/21.18%/17.90%
AM3D	pseudo point cloud	**32.23%/21.09%/17.26%**	**43.75%/28.39%/23.87%**

The Linemod dataset [28] and YCB-Video dataset are mainly include data of indoors objects, which are close to the camera. So yaw angle, roll angle and pitch angle of objects cannot be ignored. The average distance (ADD) metric is used in dataset LINEMOD and YCB-Video. Occlusion LINEMOD [29], Truncation LINEMOD is derived from the dataset LINEMOD for occlusion and truncation situation. The ADD computes the mean of the pairwise distances between the 3D model points transformed according to the ground truth transformation matrix and the estimated transformation matrix, However, the ADD is not accurate enough for the evaluation of symmetrical objects. So the metric ADD-S is proposed to evaluate symmetrical objects effectively and accurately.

The popular algorithms on indoor datasets are algorithms based single-stage 2D detection and pnp. The pnp based algorithm performs well on the LINEMOD and Occulusion LINEMOD, but it is unable to achieve single-stage, which have disadvantages in real-time performance and calculation volume. The one of future work is to improve the running speed of the pnp based algorithm under the premise of ensuring accuracy. We also show the performance of some algorithms on LINEMOD and Occulusion LINEMOD in Table 3.

Table 3. Performance of representative algorithms on (Occulusion) Linemod dataset

Algorithm	**Category**	**2D projection (Linemod)**	**ADD(-S) (Linemod)**	**2D projection (Occulusion Linemod)**	**ADD(-S) (Occulusion Linemod)**
Tekin	single-stage 2D detection	90.37%	55.95	6.16%	6.42%
PoseCNN		/	/	17.2%	24.9%
Seg-driven	pnp	/	/	44.9%	27%
PVNet		**99%**	**86.27%**	**61.06**	**40.77**

4. Summary

This paper briefly reviews the state-of-the-art of algorithms on monocular 3D object detection. We first introduce the advantages and disadvantages of monocular camera compared to other sensors. Then we summarize the algorithms for monocular 3D object detection into six types, and introduce contributions and limitations of each type of algorithm based on deep learning. Lastly, We compare the performance of different algorithms on different datasets and present directions of future work.

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), Las Vegas, NV, USA, 2016, pp. 2147–2156.

[2] T. He and S. Soatto, “Mono3D++: Monocular 3D Vehicle Detection with Two-Scale 3D Hypotheses and Task Priors,” arXiv:1901.03446 [cs], Jan. 2019.

[3] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teulière, and T. Chateau, “Deep MANTA: A Coarse-to-fine Many-Task Network for joint 2D and 3D vehicle analysis from monocular image,” arXiv:1703.07570 [cs], Mar. 2017.

[4] F. Manhardt, W. Kehl, and A. Gaidon, “ROI-10D: Monocular Lifting of 2D Detection to 6D Pose and Metric Shape,” arXiv:1812.02781 [cs], Dec. 2018.

[5] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep ordinal regression network for monocular depth estimation. In CVPR, pages 2002–2011, 2018. 2, 5, 7, 12

[6] C. Godard, O. Mac Aodha, and G. J. Brostow, “Unsupervised Monocular Depth Estimation with Left-Right Consistency,” arXiv:1609.03677 [cs, stat], Sep. 2016.

[7] Xu B, Chen Z. Multi-level fusion based 3d object detection from monocular images[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 2345-2353.

[8] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, “3D Bounding Box Estimation Using Deep Learning and Geometry,” arXiv:1612.00496 [cs], Dec. 2016.

[9] Z. Qin, J. Wang, and Y. Lu, “MonoGRNet: A Geometric Reasoning Network for Monocular 3D Object Localization,” arXiv:1811.10247 [cs], Nov. 2018.

[10] T. Roddick, A. Kendall, and R. Cipolla, “Orthographic Feature Transform for Monocular 3D Object Detection,” arXiv:1811.08188 [cs], Nov. 2018.

[11] A. Kendall, M. Grimes, and R. Cipolla, “PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization,” in 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 2938–2946. [12]

[12] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, “PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes,” arXiv:1711.00199 [cs], Nov. 2017.

[13] B. Li, W. Ouyang, L. Sheng, X. Zeng, and X. Wang, “GS3D: An Efficient 3D Object Detection Framework for Autonomous Driving,” arXiv:1903.10955 [cs], Mar. 2019.

[14] L. Liu, J. Lu, C. Xu, Q. Tian, and J. Zhou, “Deep Fitting Degree Scoring Network for Monocular 3D Object Detection,” arXiv:1904.12681 [cs], Apr. 2019.

[15] G. Brazil and X. Liu, “M3D-RPN: Monocular 3D Region Proposal Network for Object Detection,” arXiv:1907.06038 [cs], Aug. 2019.

[16] V. Lepetit, F. Moreno-Noguer, and P. Fua, “EPnP: An Accurate O(n) Solution to the PnP Problem,” International Journal of Computer Vision, vol. 81, no. 2, pp. 155–166, Feb. 2009.

[17] B. Tekin, S. N. Sinha, and P. Fua, “Real-Time Seamless Single Shot 6D Object Pose Prediction,” arXiv:1711.08848 [cs], Nov. 2017.

[18] Y. Hu, J. Hugonot, P. Fua, and M. Salzmann, “Segmentation-Driven 6D Object Pose Estimation,” p. 10.

[19] S. Peng, Y. Liu, Q. Huang, H. Bao, and X. Zhou, “PVNet: Pixel-wise Voting Network for 6DoF Pose Estimation,” arXiv:1812.11788 [cs], Dec. 2018.

[20] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation,” arXiv:1612.00593 [cs], Dec. 2016. [22] B. Tekin, S. N. Sinha, and P. Fua, “Real-Time Seamless Single Shot 6D Object Pose Prediction,” arXiv:1711.08848 [cs], Nov. 2017.

[21] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space,” arXiv:1706.02413 [cs], Jun. 2017.

[22] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, “Frustum PointNets for 3D Object Detection From RGB-D Data,” p. 10.

[23] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, “Pseudo-LiDAR from Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving,” arXiv:1812.07179 [cs], Dec. 2018.

[24] KU J, PON A D, WASLANDER S L. Monocular 3D Object Detection Leveraging Accurate Proposals and Shape Reconstruction[J]. arXiv:1904.01690 [cs], 2019.

[25] X. Ma, Z. Wang, H. Li, P. Zhang, X. Fan, and W. Ouyang, “Accurate Monocular Object Detection via Color-Embedded 3D Reconstruction for Autonomous Driving,” arXiv:1903.11444 [cs], Mar. 2019.

[26] 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, Providence, RI, 2012, pp. 3354–3361.

[27] X. Song et al., “ApolloCar3D: A Large 3D Car Instance Understanding Benchmark for Autonomous Driving,” arXiv:1811.12222 [cs], Nov. 2018.

[28] S. Hinterstoisser et al., “Model Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes,” in Computer Vision – ACCV 2012, vol. 7724, K. M. Lee, Y. Matsushita, J. M. Rehg, and Z. Hu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 548–562.

[29] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6d object pose estimation using 3d object coordinates. In ECCV, 2014. 2, 3, 6