

BOOSTING MONOCULAR 3D OBJECT DETECTION THROUGH DEPTH PROBABILISTIC OPTIMIZATION

A PREPRINT

 **Tingyu Zhang***

College of Computer Science and Technology
Jilin University
Chang Chun, Jilin
zhangty21@mails.jlu.edu.cn

 **Jian Wang**

College of Computer Science and Technology
Jilin University
Chang Chun, Jilin
wangjian591@jlu.edu.cn

March 8, 2024

ABSTRACT

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Keywords First keyword · Second keyword · More

1 Introduction

DID-M3D employs RoI-based techniques to estimate the depth of 3D bounding box center by partitioning it into two components: the visual depth, which represents the distance to the object’s surface, and the attribute depth, which measures the distance from the surface to the object’s center. To ascertain the visual depth, a depth completion network is utilized to produce a dense pixel depth map. Subsequently, the RoI is divided into a 7×7 grid, and the visual depth for each grid cell is computed using the RoIAlign operation. Although RoI-based methods typically outperform non-RoI counterparts, DID-M3D’s performance is inferior to that of Monocon. Upon thorough examination, two primary reasons for this discrepancy were identified. First, RoI-based methods are susceptible to background noise. For instance, in depth estimation tasks, pixel depths near the center of an object are more precise than those at the edges, as illustrated in Figure 2.

RoI-based methods split the RoI into different parts, and use some interpolation methods to get features of each parts. The features of each part are encoded by network and integrated into a unified result, e.g. RoI Warp, RoI Align. However, the RoI-based methods suffer from the background noise. In depth completion, the pixels near the instance center are generate more accurate depth than distant pixels, which is depicted in Fig. ?? . And the methods using depth completion outcome as an extra input suffer from this. We first divide the RoI into 7×7 grids, and use RoI Align to obtain the corresponding depth of each grid. Because we let each grid to estimate the depth, and let the weighted sum of the depth to become the final depth of RoI.

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.



Figure 1: None

2 Method

2.1 Problem Definition

In the realm of monocular 3D object detection, the principal input comprises the RGB image I . The overarching objective is to discern crucial 3D bounding box properties, specifically the 3D center coordinates x_c, y_c, z_c , the 3D dimensions l, w, h and the orientation angle θ . A pivotal component in this process is the projection matrix P , elucidated in Eq (1).

$$P = \begin{pmatrix} f & 0 & c_u & -fb_x \\ 0 & f & c_v & -fb_y \\ 0 & 0 & 1 & -fb_z \end{pmatrix} \quad (1)$$

where f is the focal length, while c_u and c_v signify the vertical and horizontal positions of the camera in the image. Additionally, b_x, b_y and b_z represent the baseline relative to the reference camera, with non-zero values in the KITTI dataset and zero values in the nuScenes dataset.

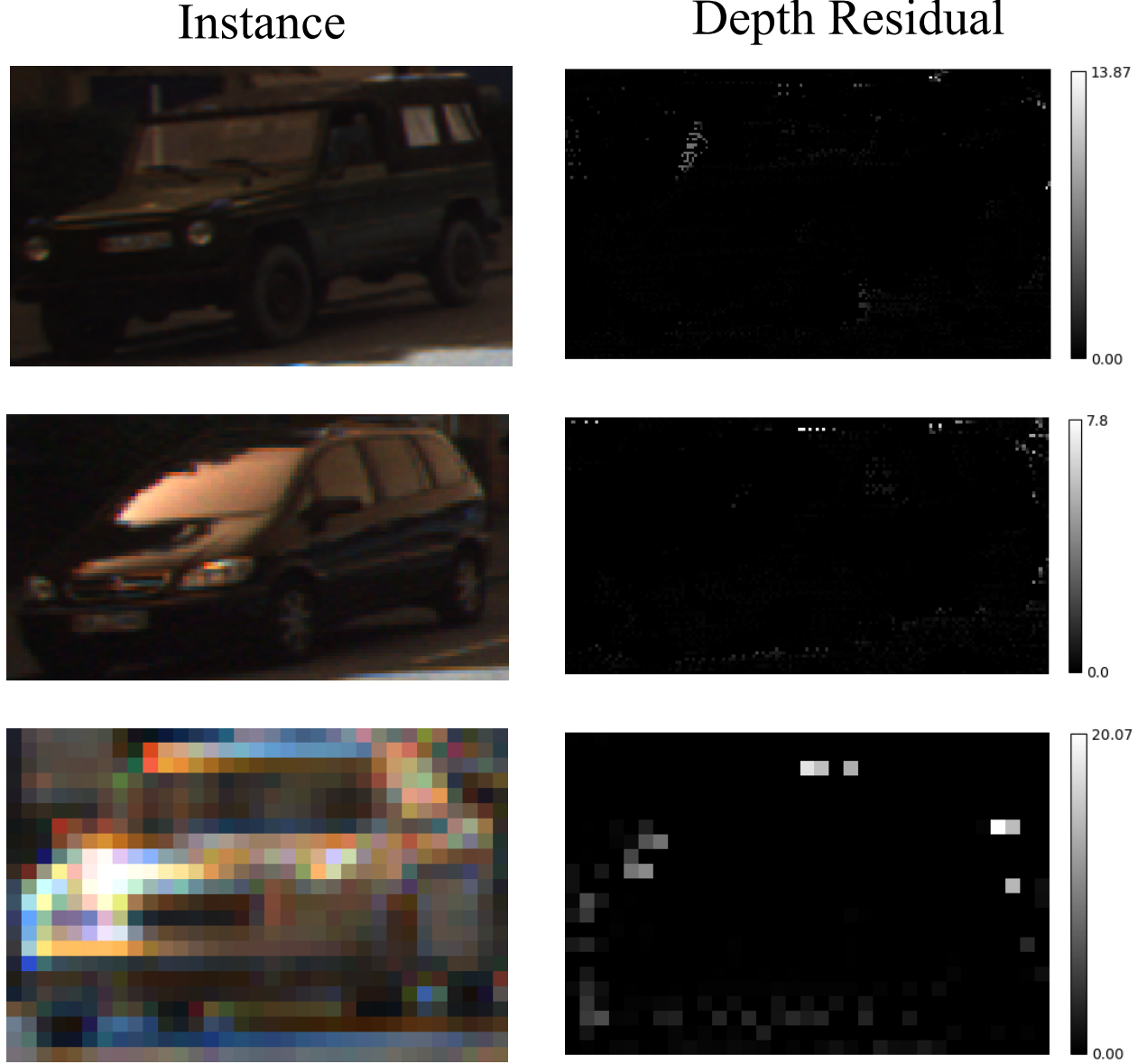


Figure 2: Instance and its depth residual. The depth residual of an instance is determined by calculating the absolute difference between the predicted depth from the depth completion network and the corresponding ground truth. The left column displays the raw image of the instance, while the right column depicts the depth residual. In the latter, a lighter color indicates a greater depth residual.

In monocular setting, ascertaining the 3D center position presents a challenging task, primarily due to the substantial variability in 3D center scale. Consequently, numerous studies resort to predicting the projected 3D center on the image plane, denoted as x_{ic}, y_{ic} , along with the corresponding depth d . The recovery of the 3D bounding box center is subsequently achieved using Eq (2).

$$dx_{2d} = Px_{3d} \quad (2)$$

where P represents the projection matrix, x_{3d} denotes the 3D bounding box center $(x_c, y_c, z_c, 1)^T$, x_{2d} is the projected 3D center on the image plane $(x_{ic}, y_{ic}, 1)^T$, and d represents the depth of x_{3d} .

2.2 Overview

The schematic overview of our methodology is presented in Figure 1. Initially, the provided image I undergoes encoding by the image backbone, as expounded in Section 2.2.1. Subsequently, a 2D detection head is employed to obtain 2D properties, such as the width and height of the 2D bounding box, and generate the heatmap of the projected 3D object center, as detailed in Section 2.2.2. Utilizing the heatmap along with the predicted width and height, the projected 2D bounding box is determined. The RoI region is uniformly divided into a 7×7 grid, and RoIAlign is applied to extract features from the predicted 2D bounding box. The features within the region are then enhanced using the transformer module, elucidated in Section 2.2.3. For each RoI grid, we employ the DID-M3D approach to predict both the visual depth and attribute depth, subsequently predicting 3D properties such as dimensions and orientation angle. In the post-processing phase, a novel depth integration strategy is applied to consolidate RoI depths, considering the practical significance of probability distribution. This strategy is detailed in Section 2.2.6.

2.2.1 Image Backbone

Given an input RGB image I with dimensions $3 \times H \times W$, we employ a feature backbone $f(\cdot; \Theta)$ to calculate the feature map F with dimensions $D \times h \times w$:

$$F = f(I; \Theta) \quad (3)$$

where Θ encompasses all the learnable parameters, D represents the output feature map dimension (e.g. $D=512$), and h and w are determined by the overall sub-sampling rate s in the backbone (e.g. $s = 4$). For the sake of equitable comparisons in our experiments, we adopt the DLA-34 network as our chosen backbone.

2.2.2 2D Detection Head

Utilizing the output feature F from the backbone as input, we direct it to three detection heads. Each detection head comprises a sequence of operations: a 2D convolution, Rectified Linear Unit (ReLU), and another 2D convolution. Specifically, the first detection head is responsible for predicting the heatmap $heat_{3d}$ of the projected 3D object center. The process of heatmap generation aligns with the operations outlined in CenterNet. The second detection head focuses on predicting the offset $\Delta x, \Delta y$ between the projected 3D object center and the center of the 2D bounding box. Finally, the third detection head is tasked with predicting the width w_{2d} and height h_{2d} of the 2D bounding box.

2.2.3 RoI Refinement Module

During the training phase, the ground truth projected 3D object center $(x_{c_{gt}}, y_{c_{gt}})$, predicted offset $\Delta x, \Delta y$, as well as the predicted width w_{2d} and height h_{2d} of the 2D bounding box are employed to compute the RoI using Eq (4).

$$RoI = (x_{c_{gt}} - w_{2d}/2, y_{c_{gt}} - h_{2d}/2, x_{c_{gt}} + w_{2d}/2, y_{c_{gt}} + h_{2d}/2) \quad (4)$$

The RoI is defined by the coordinates of the top-left and bottom-right corners. Subsequently, RoI Align is applied to uniformly partition the RoI into 7×7 grids. For each grid, the DID-M3D approach is employed to infer both the visual depth and attribute depth, with their definitions aligning with those in DID-M3D. Specially, LiDAR points are initially projected onto the image plane to generate a sparse depth map, followed by the utilization of a depth completion network to obtain a dense depth map. The ground truth 2D bounding box label defines the actual RoI region, and RoI Align is then used to acquire the visual depth label for each grid. The attribute depth of each grid is computed as the difference between the depth of the projected 3D object center and the visual depth.

In the inference phase, the top-50 positions in the heatmap are chosen to represent the predicted projected 3D object center. Additionally, the predicted offset and 2D dimensions are combined to generate the RoI.

In the context of DID-M3D, the estimation of grid depth is treated as an independent process for each grid. However, considering that each grid represents a distinct part of the object, it becomes natural to incorporate the relationships among grids. Inspired by the common use of self-attention in transformer encoders to analyze relationships between inputs, we employ a transformer encoder to enhance the features of grids belonging to the same object. Specifically, denoting the grid features as $\mathcal{G} \in \mathbb{R}^{49 \times C}$, we utilize these features as the query, key, and value inputs for the transformer encoder, resulting in the generation of enhanced Region of Interest (RoI) features, denoted as F_{RoI} . Subsequently, these enhanced RoI features F_{RoI} are fed into the 3D detection head to obtain additional properties of the 3D bounding box.

2.2.4 3D Detection Head

For each set of RoI (RoI) features, we employ six distinct detection heads to predict various properties. These include the 3D dimensions offset, denoted as dim_{3d} , in comparison to the mean size of each class; the offset $offset_{3d}$ representing the quantization error-induced displacement between the projected 3D object center and the corresponding pixel; the

visual depth d_{vis} and its associated uncertainty σ_{vis} ; the attribute depth d_{att} and its corresponding uncertainty σ_{att} ; and the orientation angle θ .

The detection heads responsible for predicting dim_{3d} , $offset_{3d}$, and θ consist of a sequence comprising one 2D convolution, BatchNorm, Rectified Linear Unit (ReLU), and another 2D convolution. On the other hand, the detection heads tasked with predicting d_{vis} , σ_{vis} , d_{att} , and σ_{att} are composed of one 2D convolution, LeakyReLU, and another 2D convolution.

2.2.5 Loss Functions

2D Detection Component The 2D detection head produces three outputs: the heatmap $heat_{3d}$, the offset $\Delta x, \Delta y$ and the 2D dimensions (w_{2d}, h_{2d}). The loss function aligns with the approach used in CenterNet. Specifically, for $heat_{3d}$, we employ a modified focal loss, defined as follows:

$$L_{heat} = \frac{-1}{N} \sum_{heat_{3d}} \begin{cases} (1 - heat_{3d})^\alpha \log(heat_{3d}) & \text{if } \hat{heat}_{3d} = 1 \\ (1 - \hat{heat}_{3d})^\beta (heat_{3d})^\alpha \log(1 - heat_{3d}) & \text{otherwise} \end{cases} \quad (5)$$

where \hat{heat}_{3d} represents the target heatmap, and N is the number of keypoints in the image. The hyper-parameters α and β for the focal loss are set to 2 and 4, respectively, in our experiments.

3D Detection Component For 3D offset $offset_{3d}$ and 3D dimensions dim_{3d} , we use L1 loss to calculate the loss, which is formulated as below.

2.2.6 Post Processing Phase

3 Introduction

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris. Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

4 Headings: first level

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula. See Section 4.

4.1 Headings: second level

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

$$\xi_{ij}(t) = P(x_t = i, x_{t+1} = j | y, v, w; \theta) = \frac{\alpha_i(t) a_{ij}^{w_t} \beta_j(t+1) b_j^{v_{t+1}}(y_{t+1})}{\sum_{i=1}^N \sum_{j=1}^N \alpha_i(t) a_{ij}^{w_t} \beta_j(t+1) b_j^{v_{t+1}}(y_{t+1})} \quad (6)$$

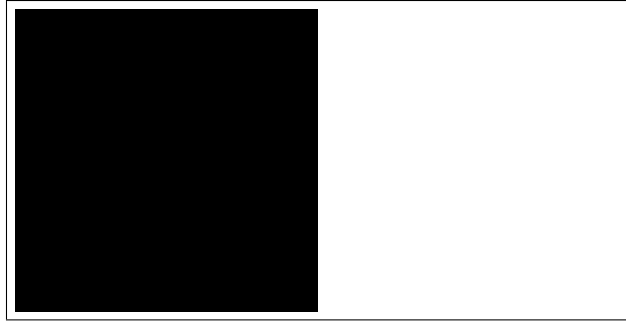


Figure 3: Sample figure caption.

4.1.1 Headings: third level

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Paragraph Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

5 Examples of citations, figures, tables, references

5.1 Citations

Citations use natbib. The documentation may be found at

<http://mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf>

Here is an example usage of the two main commands (`citet` and `citep`): Some people thought a thing [Kour and Saabne, 2014a, Hadash et al., 2018] but other people thought something else [Kour and Saabne, 2014b]. Many people have speculated that if we knew exactly why Kour and Saabne [2014b] thought this...

5.2 Figures

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetur odio sem sed wisi. See Figure 3. Here is how you add footnotes.² Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetur eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

5.3 Tables

See awesome Table 1.

The documentation for booktabs (*‘Publication quality tables in LaTeX’*) is available from:

²Sample of the first footnote.

Table 1: Sample table title

| Part | | |
|----------|-----------------|------------------------|
| Name | Description | Size (μm) |
| Dendrite | Input terminal | ~ 100 |
| Axon | Output terminal | ~ 10 |
| Soma | Cell body | up to 10^6 |

<https://www.ctan.org/pkg/booktabs>

5.4 Lists

- Lorem ipsum dolor sit amet
- consectetur adipiscing elit.
- Aliquam dignissim blandit est, in dictum tortor gravida eget. In ac rutrum magna.

References

- George Kour and Raid Saabne. Real-time segmentation of on-line handwritten arabic script. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pages 417–422. IEEE, 2014a.
- Guy Hadash, Einat Kermany, Boaz Carmeli, Ofer Lavi, George Kour, and Alon Jacovi. Estimate and replace: A novel approach to integrating deep neural networks with existing applications. *arXiv preprint arXiv:1804.09028*, 2018.
- George Kour and Raid Saabne. Fast classification of handwritten on-line arabic characters. In *Soft Computing and Pattern Recognition (SoCPaR), 2014 6th International Conference of*, pages 312–318. IEEE, 2014b. doi:10.1109/SOCPAR.2014.7008025.