Monocular 3D Object Detection with Depth from Motion
Supplementary Materials

Tai Wang[1,2] Jiangmiao Pang[2] Dahua Lin[1,2]

[1]The Chinese University of Hong Kong [2]Shanghai AI Laboratory
{wt019,dhlin}@ie.cuhk.edu.hk, pangjiangmiao@gmail.com

1 Supplementary Results

1.1 Detection Performance on the Test Set

Due to the lack of pose information on the test set, we do not provide related results in the main paper. Here we show the results of our pose-free version in Tab. 1. We can observe conclusions similar to those on the validation set. Our method shows obvious superiority over previous methods even without precise pose information. Furthermore, we can expect a more significant improvement if ego-motion is available. We will also attempt to extend our method to other datasets that satisfy this requirement, such as nuScenes and Waymo.

Table 1. AP_{40} results on the KITTI test set.

Methods	Venue	AP_{3D} IoU$\geq$ 0.7 (%)			AP_{BEV} IoU$\geq$ 0.7 (%)		
		Easy	Mod.	Hard	Easy	Mod.	Hard
MonoDIS [21]	ICCV 2019	10.37	7.94	6.40	17.23	13.19	11.12
M3D-RPN [1]	ICCV 2019	14.76	9.71	7.42	21.02	13.67	10.23
D4LCN [5]	CVPR 2020	16.65	11.72	9.51	22.51	16.02	12.55
MonoPair [4]	CVPR 2020	13.04	9.99	8.65	19.28	14.83	12.89
SMOKE [12]	CVPRW 2020	14.03	9.76	7.84	20.83	14.49	12.75
PatchNet [15]	ECCV 2020	15.68	11.12	10.17	22.97	16.86	14.97
RTM3D [10]	ECCV 2020	14.41	10.34	8.77	19.17	14.20	11.99
IAFA [26]	ECCV 2020	17.81	12.01	10.61	25.88	17.88	15.35
MoVi3D [20]	ECCV 2020	15.19	10.90	9.26	22.76	17.03	14.85
MonoDLE [16]	CVPR 2021	17.23	12.26	10.29	24.79	18.89	16.00
CaDDN [17]	CVPR 2021	19.17	13.41	11.46	27.94	18.91	17.19
MonoFlex [25]	CVPR 2021	19.94	13.89	12.07	28.23	19.75	16.89
MonoRCNN [19]	ICCV 2021	18.36	12.65	10.03	25.48	18.11	14.10
GUPNet [14]	ICCV 2021	20.11	14.20	11.77	-	-	-
DFR-Net [27]	ICCV 2021	19.40	13.63	10.35	28.17	19.17	14.84
Kinematic3D [2]	ECCV 2020	19.07	12.72	9.17	26.69	17.52	13.10
DfM w/o pose	-	**22.94**	**16.82**	**14.65**	**31.71**	**22.89**	**19.97**

1.2 Detection Performance of Other Classes

Considering the limited samples of pedestrians and cyclists on KITTI, its performance is empirically unstable. So we mainly compare the detection performance

of cars previously. Here, we also provide related results in Tab. 2 for reference. It can be seen that our method also achieves competitive results, especially on the detection of cyclists. For the detection of pedestrians, our method is only a little inferior to GUPNet [14]. We suspect the reason is that the detection of small objects can be hard for BEV-based methods. From this perspective, our method achieves better performance than CaDDN, which follows a similar detection pipeline.

Table 2. AP_{40} results of other classes on the KITTI test set.

Methods	Venue	Ped@AP_{3D} IoU$\geq$ 0.5 (%)			Cyc@AP_{3D} IoU$\geq$ 0.5 (%)		
		Easy	Mod.	Hard	Easy	Mod.	Hard
M3D-RPN [1]	ICCV 2019	4.92	3.48	2.94	0.94	0.65	0.47
D4LCN [5]	CVPR 2020	4.55	3.42	2.83	2.45	1.67	1.36
MonoPair [4]	CVPR 2020	10.02	6.68	5.53	3.79	2.12	1.83
MoVi3D [20]	ECCV 2020	8.99	5.44	4.57	1.08	0.63	0.70
MonoDLE [16]	CVPR 2021	9.64	6.55	5.44	4.59	2.66	2.45
CaDDN [17]	CVPR 2021	12.87	8.14	6.76	7.00	3.41	3.30
MonoFlex [25]	CVPR 2021	9.43	6.31	5.26	4.17	2.35	2.04
GUPNet [14]	ICCV 2021	**14.72**	**9.53**	**7.87**	4.18	2.65	2.09
DFR-Net [27]	ICCV 2021	6.09	3.62	3.39	5.69	3.58	3.10
Kinematic3D [2]	ECCV 2020	-	-	-	-	-	-
DfM w/o pose	-	13.70	8.71	7.32	**8.98**	**5.75**	**4.88**

1.3 Latency of Constructing Cost Volume

In the main paper, we mentioned that although cost volume construction becomes more complicated than that in the binocular system, it is overall achieved with matrix multiplication. The additional complexity increases the latency of this process from 0.003s to 0.012s, which can be ignored for the overall inference latency of 0.32s. Although our framework does not achieve real-time efficiency, it has performed better than similar baselines such as CaDDN (0.63s) and Pseudo-LiDAR based methods (about 0.4s). In addition, we can reduce the number of candidate depth levels to optimize the network efficiency while affecting little performance. We will also improve our framework in this aspect in the future.

Table 3. Ablation study for location-aware monocular compensation.

Methods	AP_{3D} IOU$\geq$ 0.7			AP_{BEV} IOU$\geq$ 0.7		
	Easy	Mod.	Hard	Easy	Mod.	Hard
stereo baseline	21.47	15.32	13.83	29.22	21.22	19.51
w/ shared weights	22.92	15.99	13.85	31.31	23.00	20.22
group-wise fusion	23.49	16.52	14.38	33.00	23.91	21.06
point-wise fusion	26.61	18.82	16.47	36.16	26.09	23.17

1.4 Supplementary Ablation Studies

Alternative Monocular Compensation Methods We also attempt alternative methods to fuse monocular and stereo features. The final version in the main paper is both interpretable and effective. To have a more comprehensive comparison, we also show the results of other alternative designs for monocular compensation in Tab. 3. First, we use a simple convolution layer to directly compress these two feature volumes to one, *i.e.*, compress from $2D$ channels to D. This implementation is simple while it essentially uses shared weights to aggregate these two volumes across the entire scene. As analyzed in the paper, we need to fuse adaptively because different locations can rely on monocular or stereo estimation differently. Then we attempt to use group-wise convolution to achieve this. Finally, our final version, generating a point-wise weight first and then using it to guide the fusion, is the most effective design. It is also in line with our theoretical analysis.

Table 4. Ablation study for depth loss design.

Methods	AP_{3D} IOU$\geq$ 0.7			AP_{BEV} IOU$\geq$ 0.7		
	Easy	Mod.	Hard	Easy	Mod.	Hard
Mono Only w/ CE	20.06	15.30	14.05	27.84	21.78	19.96
Stereo Only w/ CE	21.47	15.32	13.83	29.22	21.22	19.51
Mono+Stereo w/ CE	26.61	18.82	16.47	36.16	26.09	23.17
focal w/ gamma=2	27.27	18.76	16.55	35.29	25.08	22.11
balanced w/ fg:bg=5:1	27.40	19.11	16.58	36.28	26.18	23.09
balanced + focal	29.27	20.22	17.46	38.60	27.13	24.05

Table 5. Depth estimation errors when using different loss designs. Err. Med. denotes the average median of depth errors and other metrics evaluate the ratio of points with errors larger than a specific threshold. Foreground (Fg in the table) metrics are evaluated by averaging object-level results. Objects with less than 5 ground-truth LiDAR points are ignored.

Methods	Err. Med.↓	>0.2m↓	>0.4m↓	>0.8m↓	>1.6m↓
	Fg/All (m)	Fg/All (%)	Fg/All (%)	Fg/All (%)	Fg/All (%)
Mono Only w/ CE	5.86/1.15	91.2/75.6	83.9/64.4	74.6/53.1	65.8/43.3
Stereo Only w/ CE	3.33/0.58	88.6/68.5	79.3/52.4	66.6/36.0	51.9/23.0
Mono+Stereo w/ CE	2.60/0.48	86.3/66.8	75.0/50.2	60.0/33.4	43.9/20.7
focal w/ gamma=2	2.59/0.48	86.3/67.0	75.2/50.1	60.4/33.2	44.2/20.6
balanced w/ fg:bg=5:1	2.12/0.51	83.2/67.7	70.2/51.3	53.6/34.6	35.7/21.7
balanced + focal	2.09/0.50	82.8/67.2	69.7/50.9	53.1/34.2	35.4/21.3

Design of Depth Loss Our baseline uses cross-entropy loss for depth supervision. Since our target is 3D object detection, we should pay more attention to foreground points. Therefore, following CaDDN [17], we use focal design and balanced weights to facilitate the depth estimation from this aspect. We show their

effectiveness in Tab. 4 and 5. To have a more intuitive comparison, we also show related results of monocular and stereo only baselines. We can see these designs tailored to depth contribute a lot to the final performance improvement, which further shows the crucial role of depth estimation in monocular 3D detection.

Table 6. Our baseline performs much worse than its binocular counterpart. The key is the accuracy of depth distribution.

Methods	AP_{3D} IOU$\geq$ 0.7			AP_{BEV} IOU$\geq$ 0.7		
	Easy	Mod.	Hard	Easy	Mod.	Hard
Binocular Baseline	80.62	61.88	54.92	90.26	73.63	66.24
w/ gt depth dist.	85.41	70.07	62.96	93.82	82.24	74.89
DfM Baseline	17.41	12.93	11.60	24.78	18.21	16.06
w/ gt depth dist.	76.70	63.01	55.74	87.47	76.62	69.09

1.5 Oracle Analysis for Baseline Model

When we build our baseline framework at the beginning (w/o data augmentation and monocular compensation), it turns out that the detection performance drops precipitously compared to the binocular baseline counterpart. However, if we replace the predicted depth distribution $\hat{D}_P$ with its target D_P, our baseline can be directly lifted to a level comparable with the binocular case. Although this assumption is a little idealistic, it still indicates that the key problem of this large gap is the accuracy of depth estimation. Therefore, we focus on improving the depth-from-motion component in the main paper and propose two effective designs.

1.6 Qualitative Results

We show detection results qualitatively in Fig. 1. For each sample, we visualize 2D detection and 3D detection results from the front view on the first two rows and plot 3D detection results in the perspective view on the third row. For the perspective view, we also reconstruct the point clouds with our estimated depth and paint them with corresponding colors. Please see qualitative results for 3D detection from consecutive frames in the supplementary demo video.

2 Theoretical Analysis for General Two-View Cases

We have discussed the geometry relationship in different two-view cases in the main paper, especially the two simplest cases. Although the cases with ego-motion and object motions are not important for the basic conclusion and our technical design, we still provide a basic analysis here for integration. It can also provide guidance for future work in this direction.

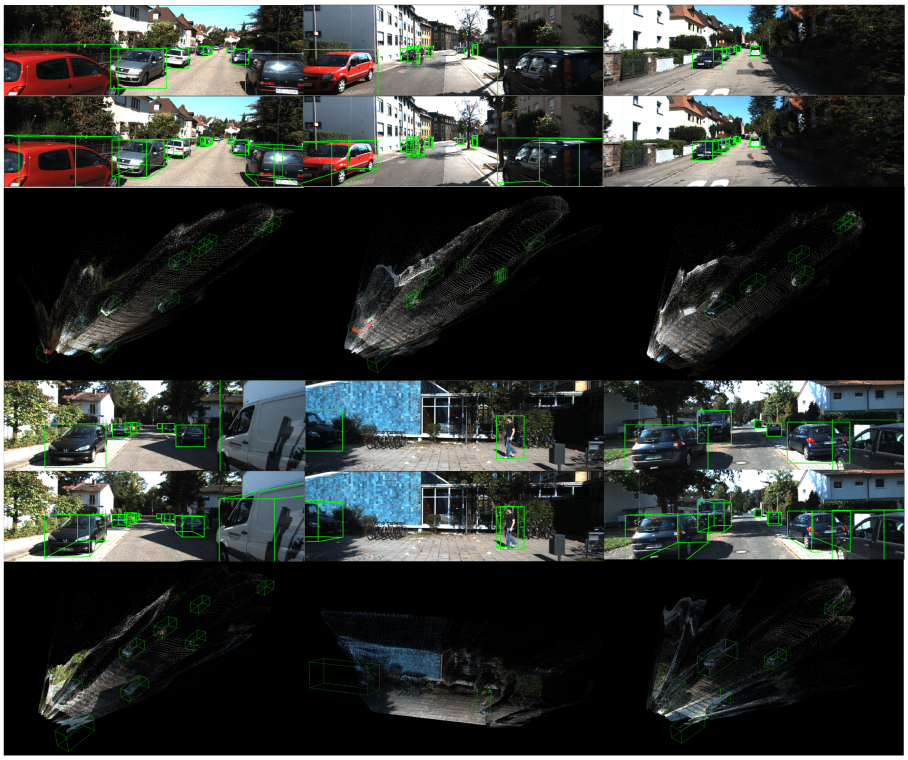

Fig. 1. Qualitative detection results from the front view and perspective view.

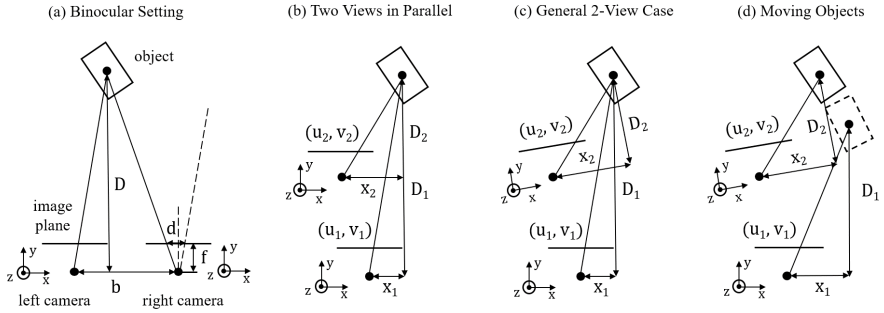

Fig. 2. Multi-view geometry for object depth estimation in the (a) binocular, (b) parallel two-view, (c) general two-view system and (d) that for moving objects.

2.1 General Two-View Case

Following the basic analysis for the binocular system and two-parallel-view case, we extend the geometry analysis to the most general one without considering the motions of target objects: The pose transformation between two views consists of both translation and rotation (Fig. 2-(c)). Similar to the analysis for two parallel views, what we have are two projection relationships and the pose transformation:

$$\begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} D_1 = \begin{pmatrix} f & 0 & c_u \\ 0 & f & c_v \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ D_1 \end{pmatrix}, \tag{1}$$

$$\begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} D_2 = \begin{pmatrix} f & 0 & c_u \\ 0 & f & c_v \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_2 \\ y_2 \\ D_2 \end{pmatrix}, \tag{2}$$

$$\begin{pmatrix} x_2 \\ y_2 \\ D_2 \end{pmatrix} = R_{3\times 3} \begin{pmatrix} x_1 \\ y_1 \\ D_1 \end{pmatrix} + t_{3\times 1}. \tag{3}$$

Represent x_1, x_2, y_1, y_2 with D_1, D_2, and substitute them in the transformation equation, and we can derive D_1 and D_2:

$$\begin{aligned} \Rightarrow D_2 &= (r_{31}\frac{u_1 - c_u}{f} + r_{32}\frac{v_1 - c_v}{f} + r_{33})D_1 + t_3 \\ &\triangleq A_3 D_1 + B_3, \end{aligned} \tag{4}$$

where r_{ij} denotes the i-th row, j-th column element of the rotation matrix R. Similarly, we can define:

$$r_{11}\frac{u_1 - c_u}{f} + r_{12}\frac{v_1 - c_v}{f} + r_{13} \triangleq A_1, t_1 \triangleq B_1 \tag{5}$$

$$r_{21}\frac{u_1 - c_u}{f} + r_{22}\frac{v_1 - c_v}{f} + r_{23} \triangleq A_2, t_2 \triangleq B_2 \tag{6}$$

Then:

$$D_1 = (B_1 - \frac{u_2 - c_u}{f}B_3)/(\frac{u_2 - c_u}{f}A_3 - A_1) \tag{7}$$

$$D_1 = (B_2 - \frac{v_2 - c_v}{f}B_3)/(\frac{v_2 - c_v}{f}A_3 - A_2) \tag{8}$$

When there is no rotation, setting R to the identity matrix, it can also be reduced to the case with two views in parallel.

For this most complicated relationship, we can also understand it from the previous two cases. We first re-written Eqn. 7 as follows:

$$D_1 = \frac{f(t_1 - \frac{u_2 - c_u}{f}t_3)}{\left(a_{31}a_{32}a_{33}\right)\begin{pmatrix} \frac{u_1 - c_u}{f} \\ \frac{v_1 - c_v}{f} \\ 1 \end{pmatrix}(u_2 - c_u) - \left(a_{11}a_{12}a_{13}\right)\begin{pmatrix} \frac{u_1 - c_u}{f} \\ \frac{v_1 - c_v}{f} \\ 1 \end{pmatrix} f} \tag{9}$$

The numerator is the same with Eqn. 3 in the main paper while the denominator is coupled with some rotations. Here we can also substitute $\frac{u_1-c_u}{f}$ and $\frac{v_1-c_v}{f}$ with $\frac{x_1}{D}$ and $\frac{y_1}{D}$ (both correspond to rotations).

After primarily interpreting the result, let us recap Eqn. 7, which is more clear for implementation. Here, if we would like to estimate depth directly, we need to predict u_1, u_2, v_1, and other values are constant given by the dataset. We can further transform the prediction of u_2 to $u_1 + \Delta u$ to simplify the learning target and turn to address the correspondence problem. However, the prediction of u_1 and v_1 can also be inaccurate, so we can first use ground truths (target values of u_1 and v_1) to learn Δu to observe whether it can converge or not. It turns out that even the task has been simplified a lot, it is still quite difficult from our preliminary experimental attempts. This is what we mentioned in the main paper: In this case, disparity computation involves several rotation coefficients and additional dimensions of absolute position v_1. The cumulative errors caused by the entanglement of multiple estimations make the direct derivation intractable.

2.2 Moving Objects

Up to now, all the formulation assumes the object is static. However, there are many moving objects in the open world. Next, we will discuss what will happen if we consider the moving objects.

Let us consider the 3D center of a car (Fig. 2-(d)): it can only drift (both in 3D and 2D) when the car has a translation. Rotation does not affect its 3D location and thus does not affect its 2D projection. Therefore, for object centers, the only difference in the previous relationship (Eq. 9) is just that the object translation should be added into the translation vector $t_{3\times1}$. However, this can be different for other 3D points. For example, the points on the object surface can rotate with the object's rotation, which can be hard to formulate with our current modeling.

The basic analysis shows that moving objects can involve local warping to monocular images, in contrast to global warping caused by view change. Due to the complexity of different objects' motion and the domain gap between the 3D targets and 2D inputs, it is hard to directly estimate motion from only a pair of images, not to mention involving the estimation errors in the direct computation of depth.

From the perspective of our framework in the main paper, a promising direction is to model the local warping when constructing stereo cost volume and attempt to remove this factor for stereo matching. More annotations such as complete tracklets may be required for better performance.

3 Implementation Details

In the main paper, we have introduced our overall framework and detailed our proposed two key components. This supplemental section elaborates on the

specific network architectures of other basic modules and presents the design related to auxiliary tasks except for 3D detection.

Our framework is motivated by DSGN [3] and LIGA-Stereo [7]. We will also release our code afterward for reproducing our experiments and showing these details more conveniently.

3.1 Network Architecture

2D Feature Extraction Given the input image-pair $(I_t, I_{t-\delta t})$, we use a shared 2D backbone to extract their features $(\mathcal{F}_t, \mathcal{F}_{t-\delta t})$. The backbone is a modified ResNet34 [9] with spatial pyramid pooling (SPP) [8] module following DSGN [3]. The channels of $conv2-5$ in ResNet are set to {64,128,128,128}. We append a small U-Net [18] on top to upsample these SPP features to get the full resolution $\mathcal{F}_t$ for high-quality stereo matching while use 2-layer convolution to extract the semantic feature $\mathcal{F}_{sem}$ [7]. The final number of channels are set to 32 for both $\mathcal{F}_t$ and $\mathcal{F}_{sem}$.

2D Detection Head We construct five-level FPN [11] by appending multiple stride-2 convolution layers on the SPP feature of frame t. Then we attach a 2D detection head for each level following ATSS [24]. Each position only has one anchor box and the anchor box sizes on each level are set to {32, 64, 128, 256, 512}.

2.5D Backbone After constructing the monocular and stereo cost volume, we filter each with a 3D residual block and a 3D hourglass network separately. The residual block consists of two 3D convolution layers and a skip connection as the basic block in ResNet. The 3D hourglass network downsamples the 3D feature with two stride-2 layers and then upsamples them with skip connections. We set the 3D kernel size to 3×3×3 by default. While all these operations are 3D convolutions, we call this component as 2.5D backbone because the spatial quantization is based on the 2.5D coordinates, *i.e.*, following the plane-sweep approach, which is different from the voxelization in the 3D space.

3D Backbone and 3D Head With the fused stereo feature, the depth head applies a simple 3D convolution layer followed by softmax to predict the depth distribution. We further apply the outer product to the semantic feature F_{sem} and the depth probability volume D_P, and combine it with the stereo feature for sampling the voxel features used for subsequent 3D detection. The voxel feature is then filtered with a 3D convolution layer and downsampled along the height axis. We transform this feature by merging its height and feature dimension to get the bird-eye-view (BEV) feature. A 2D hourglass network with 2-layer downsampling and upsampling is applied on top to get the input of 3D heads. Finally, we append two layers for the classification and regression branch separately and use one layer for each task: classification, direction classification and regression. We follow the rotation encoding scheme in SECOND [23], and use kernel size 3×3 by default for all the layers except the final one for direction classification.

3.2 Training Loss

In summary, we use a focal loss $\mathcal{L}_{depth}$ for depth supervision following [17], an auxiliary 2D detection loss $\mathcal{L}_{2D}$ and a 3D detection loss $\mathcal{L}_{3D}$ composed of focal loss for classification, regression L1 loss, IoU loss and direction loss for localization following [7]. To make the paper self-contained, we briefly introduce them as follows.

First, the baseline cross-entropy depth loss is:

$$\mathcal{L}_{depth} = \frac{1}{N_{gt}} \sum_{u,v} \sum_{w} \Big[- max(1 - \frac{|d^* - d(w)|}{\Delta d}, 0) log \mathcal{D}_P(u, v, w) \Big], \tag{10}$$

where N_{gt} is the number of valid pixels with depth ground truth d^*, u, v, w denotes the position in the stereo volume, Δd is the divided depth interval as in the main paper.

We upgrade it with balanced weights and focal design to make it more concentrated on foreground points. The foreground and background weight of depth estimation is set to 5 and 1, and γ is set to 2 in the focal loss. Here we regard the regions in the annotated 2D bounding boxes as foreground.

2D detection loss $\mathcal{L}_{2D}$ consists of three parts: focal loss for classification $\mathcal{L}_{2D}^{cls}$, GIoU loss for localization $\mathcal{L}_{2D}^{GIoU}$ and cross-entropy loss for centerness $\mathcal{L}_{2D}^{ct}$. The weights of them are set to 1.0, 2.0, 1.0 respectively.

3D detection loss $\mathcal{L}_{3D}$ has four components: focal loss for 3D classification $\mathcal{L}_{3D}^{cls}$, regression L1 loss $\mathcal{L}_{3D}^{reg}$ and IoU loss $\mathcal{L}_{3D}^{IoU}$ for localization, and cross-entropy loss for direction classification $\mathcal{L}_{3D}^{dir}$. Except for IoU loss, the others are devised following SECOND [23]. The IoU loss is defined as the average rotated IoU loss between the predicted boxes and ground truth boxes. The weights of them are set to 1.0, 0.5, 1.0, 0.2. In addition, we keep the original imitation loss $\mathcal{L}_{im}$ in LIGA-Stereo [7] to learn better geometric information. We keep its weight to 1.0 and obtain a little performance gain of about 1 AP in our baseline.

3.3 Training Details

Detection Range As for the detection range, we set $[2m, 59.6m]$ for Z (depth) axis, $[-30m, 30m]$ for X axis and $[-1m, 3m]$ for Y (height) axis to avoid more false positives too far away. The depth range is divided into 288 levels and the voxel size is set to $(0.2m, 0.2m, 0.2m)$.

Training Parameters For all the experiments, except ResNet backbone pre-trained on ImageNet, we trained randomly initialized networks from scratch following end-to-end manners. The network is trained using AdamW [13] optimizer, with $\beta_1 = 0.9$, $\beta_2 = 0.999$. We use 8 GPUs with 1 training sample on each to train the model for 60 epochs. The learning rate is set to 0.001 for the first 50 epochs and then reduced to 0.0001. The weight decay is set to 0.0001.

Data Augmentation As presented in the main paper, we can apply any kind of data augmentation to input images with the canonical space as the bridge. In practice, we exploit image flip and resize augmentation in turn, and the resize

range is set to $[0.95, 1.05]$. Subsequently, we fix the input image size to 320×1248 by cropping the upper part which does not contain any object. Note that we only apply the corresponding augmentation in 3D space for flip, and instead manipulate the intrinsic matrix for image rescaling and cropping.

3.4 Supplementary Details in Pose-Free DfM

View Synthesis When computing the self-supervised loss for pose learning in the pose-free DfM, the main paper mentioned that we need to synthesize the frame t with frame $t-\delta t$. Here we detail the synthesis procedure.

With the dense depth estimation $\hat{D}_t$, we can obtain a stereo grid by reprojecting the 2D grid of frame t with the intrinsic matrix. Then we warp these positions to the frame $t-\delta t$ with the predicted pose $(\mathbf{t}, \mathbf{q})$ and project them to the image plane to sample corresponding pixels. The sampled result is the expected synthesized $I_{t-\delta t\to t}$. Note that we also apply the pose-based warping in the canonical space similar to the construction of cost volume in this procedure.

Loss Formulation We also provide the specific formulation of the appearance matching loss $\mathcal{L}_p$ and the depth smoothness loss $\mathcal{L}_s$ mentioned in the main paper to make this paper self-contained:

$$\mathcal{L}_p(I_t, I_{t-\delta t\to t}) = \frac{\alpha}{2}(1 - SSIM(I_t, I_{t-\delta t\to t})) + (1-\alpha)||I_t - I_{t-\delta t\to t}|| \tag{11}$$

$$\mathcal{L}_s(\hat{D}_t) = |\delta_x \hat{D}_t| e^{-|\delta_x I_t|} + |\delta_y \hat{D}_t| e^{-|\delta_y I_t|} \tag{12}$$

$\mathcal{L}_p$ is formed by the Structural Similarity (SSIM) [22] term and the L1 pixel-wise loss term. $\mathcal{L}_s$ is used to regularize the predicted depth map $\hat{D}_t$ on texture-less low-image gradient regions (with small δ_x and δ_y). We also follow [6] on auto-mask techniques and hyper-parameter settings ($\alpha = 0.85$ and $\lambda_s = 0.001$).

4 Supplementary Video

We attach a video in the supplementary material. This video first combs out our method's general logic and specific content so that readers can understand or recap it quickly. The end shows some demo videos of the 3D detection results predicted by our model, from the perspective view and 3D view, respectively. It supplements the main paper on the qualitative results of consecutive-frame images. The video is compressed in the supplementary file. Please see the full version provided at https://github.com/Tai-Wang/Depth-from-Motion.

References

1. Brazil, G., Liu, X.: M3d-rpn: Monocular 3d region proposal network for object detection. In: IEEE International Conference on Computer Vision (2019)
2. Brazil, G., Pons-Moll, G., Liu, X., Schiele, B.: Kinematic 3d object detection in monocular video. In: Proceedings of the European Conference on Computer Vision (2020)
3. Chen, Y., Liu, S., Shen, X., Jia, J.: Dsgn: Deep stereo geometry network for 3d object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12536–12545 (2020)
4. Chen, Y., Tai, L., Sun, K., Li, M.: Monopair: Monocular 3d object detection using pairwise spatial relationships. In: IEEE Conference on Computer Vision and Pattern Recognition (2020)
5. Ding, M., Huo, Y., Yi, H., Wang, Z., Shi, J., Lu, Z., Luo, P.: Learning depth-guided convolutions for monocular 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11672–11681 (2020)
6. Guizilini, V., Ambrus, R., Pillai, S., Raventos, A., Gaidon, A.: 3d packing for self-supervised monocular depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2485–2494 (2020)
7. Guo, X., Shi, S., Wang, X., Li, H.: Liga-stereo: Learning lidar geometry aware representations for stereo-based 3d detector. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3153–3163 (2021)
8. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence **37**(9), 1904–1916 (2015)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)
10. Li, P., Zhao, H., Liu, P., Cao, F.: Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving. In: European Conference on Computer Vision (2020)
11. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (2017)
12. Liu, Z., Wu, Z., Tóth, R.: Smoke: Single-stage monocular 3d object detection via keypoint estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 996–997 (2020)
13. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
14. Lu, Y., Ma, X., Yang, L., Zhang, T., Liu, Y., Chu, Q., Yan, J., Ouyang, W.: Geometry uncertainty projection network for monocular 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)
15. Ma, X., Liu, S., Xia, Z., Zhang, H., Zeng, X., Ouyang, W.: Rethinking pseudo-lidar representation. In: European Conference on Computer Vision. pp. 311–327. Springer (2020)
16. Ma, X., Zhang, Y., Xu, D., Zhou, D., Yi, S., Li, H., Ouyang, W.: Delving into localization errors for monocular 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)
17. Reading, C., Harakeh, A., Chae, J., Waslander, S.L.: Categorical depth distributionnetwork for monocular 3d object detection. CVPR (2021)

18. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
19. Shi, X., Ye, Q., Chen, X., Chen, C., Chen, Z., Kim, T.K.: Geometry-based distance decomposition for monocular 3d object detection. In: IEEE International Conference on Computer Vision (2021)
20. Simonelli, A., Bulò, S.R., Porzi, L., Ricci, E., Kontschieder, P.: Towards generalization across depth for monocular 3d object detection. In: Proceedings of the European Conference on Computer Vision (2020)
21. Simonelli, A., Bulò, S.R.R., Porzi, L., López-Antequera, M., Kontschieder, P.: Disentangling monocular 3d object detection. In: IEEE International Conference on Computer Vision (2019)
22. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
23. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18**(10) (2018)
24. Zhang, S., Chi, C., Yao, Y., Lei, Z., Li, S.Z.: Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
25. Zhang, Y., Lu, J., Zhou, J.: Objects are different: Flexible monocular 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)
26. Zhou, D., Song, X., Dai, Y., Yin, J., Lu, F., Liao, M., Fang, J., Zhang, L.: Iafa: Instance-aware feature aggregation for 3d object detection from a single image. In: Proceedings of the Asian Conference on Computer Vision (2020)
27. Zou, Z., Ye, X., Du, L., Cheng, X., Tan, X., Zhang, L., Feng, J., Xue, X., Ding, E.: The devil is in the task: Exploiting reciprocal appearance-localization features for monocular 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)