Learning to Segment Rigid Motions from Two Frames:

SUPPLEMENTARY MATERIALS

1. Rigidity cost maps

In Sec. 3.2, we briefly motivated the design choice of using rigidity cost-maps inputs, and we expand the particular cost functions here. Given motion correspondences $(\mathbf{p_0}, \mathbf{p_1}) \in R^2$, camera intrinsics $(\mathbf{K_0}, \mathbf{K_1})$, and camera motion $\mathbf{R_c} \in SO(3)$, $\mathbf{T_c} \in R^3$, we construct four geometric motion cost maps that are tailored to particular motion configurations, including 1) an epipolar cost, 2) a homography cost, 3) a 3D P+P cost, and 4) a depth contrast cost.

1) Epipolar costs are applied to detect general moving objects, computed as the classic Sampson error [7] per-pixel. We include it here for completeness:

$$c_{\text{epi}} = \frac{(\mathbf{\tilde{p}_1}^T \mathbf{F} \mathbf{\tilde{p}_0})^2}{(\mathbf{F}\mathbf{\tilde{p}_0})_1^2 + (\mathbf{F}\mathbf{\tilde{p}_0})_2^2 + (\mathbf{F}^T\mathbf{\tilde{p}_1})_1^2 + (\mathbf{F}^T\mathbf{\tilde{p}_1})_2^2 + \epsilon}, \tag{1}$$

where $\mathbf{F} = \mathbf{K_1}^{-T}\mathbf{R}[\mathbf{t}]_\times \mathbf{K_0}^{-1}$ is the fundamental matrix and $(\mathbf{\tilde{p}_0}, \mathbf{\tilde{p}_1})$ are motion correspondences in the homogeneous coordinates. $\epsilon = 10^{-9}$ is a constant value added for numerical stability.

2) Homography costs are applied to deal with motion degeneracies in epipolar geometry [16], when it becomes difficult to estimate camera translation, but not rotation [2]. A visual comparison between epipolar costs and homography costs can be found in Fig. 3. The homography cost is implemented as per-pixel symmetric transfer error [4] with regard to the rotational homography, $\mathbf{H_R} = \mathbf{K_0}\mathbf{R_c}\mathbf{K_1}^{-1}$,

$$c_{\text{hom}} = d(\mathbf{\tilde{p}_0}, \mathbf{H_R}\mathbf{\tilde{p}_1})^2 + d(\mathbf{\tilde{p}_1}, \mathbf{H_R}^{-1}\mathbf{\tilde{p}_0})^2, \tag{2}$$

where $d(\cdot, \cdot)$ is the Euclidian image distance between two points.

3) 3D P+P costs are applied to detect the coplanar motion, where points are moving along the epipolar line (not detectable by the epipolar costs, as analyzed in Sec. 3.1. Our 3D P+P cost is extended from the 2D residual error of [1],

$$c_{\text{3D}} = ||\mathbf{\tilde{T}_{sf}}|| \cdot |\sin\beta|, \tag{3}$$

where $\beta = |\angle(\mathbf{\tilde{T}_{sf}}, -\mathbf{T_c})|$ is the measured angle between the normalized scene flow $\mathbf{\tilde{T}_{sf}}$ (as computed through optical expansion using the method of [18]) and negative camera translation $-\mathbf{T_c}$, capped to $\frac{\pi}{2}$. A visual comparison is shown in Fig. 4.

4) Depth contrast costs are applied to address the colinear motion ambiguity, where points are moving opposite to the camera translation direction in 3D, and therefore not detectable by the above costs, as shown in Fig. 5. The depth contrast cost is implemented as:

$$c_{\text{depth}} = |\log(\frac{Z^{\text{flow}}}{\gamma Z^{\text{prior}}})|, \tag{4}$$

where the flow-triangulated depth Z_0^{flow} can be computed efficiently using midpoint or DLT triangulation algorithm [7], the monocular depth prior Z_0^{prior} can be represented by a data-driven monocular depth network [6], and the scale factor γ that globally aligns Z_0^{prior} to Z_0^{flow} can be determined by the ratio of their medians. A visual comparison between the flow-triangulated depth and monocular depth prior is shown in Fig. 6.

Table 1: Details for training and inference. C: FlythingChairs [3]. T: FlythingThings [11]. SF: SceneFlow [11]. V: VIPER [13]. The optical flow network is trained sequentially on C, T, and C+SF+V.

Parameter	Value
Optical flow	
Network architecture	VCN [17]
Optimizer	Adam [9]
Learning rate	1×10^{-3}+One-cycle
Batch size / iterations on C	16 image pairs / 70k
Batch size / iterations on T	16 image pairs / 70k
Batch size / iterations on C+SF+V	12 image pairs / 70k
Optical expansion	
Network backbone	U-Net [14, 18]
Optimizer	Adam [9]
Learning rate	1×10^{-3}+One-cycle
Batch size / iterations on SF	12 image pairs / 70k
Rigid motion segmentation	
Network backbone	U-Net / DLA-34 [14, 19, 20]
Optimizer	Adam [9]
Learning rate	5×10^{-4}+One-cycle
Batch size / iterations on SF	12 image pairs / 70k
Rigid body scene flow	
# data poitns / iterations for RANSAC	3k / 1k
# iterations of LM optimization	20
Average time on KITTI / pair	1.3s

2. Training details

The details for training optical flow, optical expansion and rigid motion segmentaion networks are shown in Tab. 1.

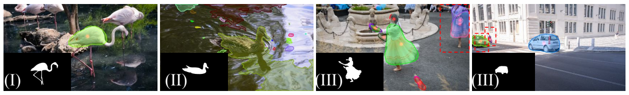


Figure 1: Failure cases with colored centers and mask predictions. GT annotations are shown at the bottom left. Red boxes in case (III) indicate moving objects predicted by ours but not labelled by DAVIS.

3. Details of rigid body scene flow

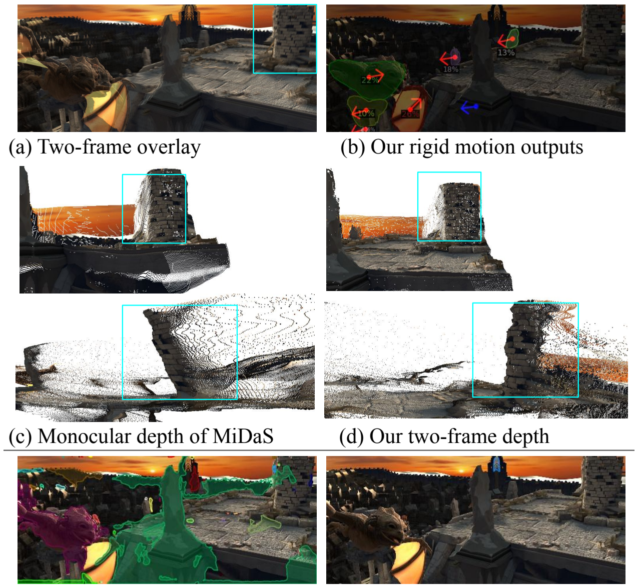
In Sec. 3.2, we describe rigid body scene flow that (1) fits 3D rigid motions per rigid body, and (2) updates depth as well as flow measurements. More details are provided here.

Overall, our goal is to select *high-quality flow correspondences* for model fitting, and update the rigid bodies with *large enough motion*. To do so, we first define "valid pixels" as pixels with flow confidence (in range 0-1, estimated by VCN [17]) greater than 0.5. During fitting, we use flow correspondences of valid pixels from each rigid motion mask to fit an essential matrix through a least median of squares estimator [15]. Then, each essential matrix is decomposed to four rotations and up-to-scale translations, where only one is feasible through cheirality check [7]. To determine the scale of translation, we triangulate flow correspondences at valid pixels and align it with the initial depth input by a scale factor through RANSAC [5]. To take advantage of accurate depth estimation in the stereo case, we refine the estimated rigid transformations by solving a Perspective-n-Point problem given first frame depth and flow that minimizes re-projection errors with Levenberg–Marquardt algorithm [7].

Finally, we update depth and flow estimations according to the estimated 3D rigid motions. Rigid bodies whose average parallax flow magnitude (defined as "rectified" optical flow after rotation removal in Sec.3.1) is lower than 4px, or has fewer than 30% valid pixels are not updated.

4. Failure cases

Although our method outperforms prior art on the challenging KITTI and Sintel datasets, we observe three types of failures when applied to DAVIS (Fig. 1): (I) Our method successfully detects centers of moving bodies, but fails to segment the exact boundary of non-rigidly deforming objects. (II) Our method fails to estimate camera motion when the background is dominated by deforming particles, such as water and smoke. (III) Our method estimates rigidity from two-frames. Over such small time scales, parts of moving objects might be regarded as rigid - e.g., the foot of a dancer that remains still for two frames may be grouped with the rigid background. Moreover, we segment *all* rigidly-moving objects, while DAVIS focuses on "visually salient" objects, treating others as false positives. Future work could improve results by processing more than 2 frames or using appearance cues.



(a) Two-frame overlay (b) Our rigid motion outputs

(c) Monocular depth of MiDaS (d) Our two-frame depth

(e) Foreground segmentation of RTN (f) Segmentation of Mask R-CNN

Figure 2: Results on Sintel sequence temple_2, frame 17-18. (a)-(b) Our method segments rigid motions and fits 3D rigid transformations over two frames. The blue and red arrows indicate the estimated motion of the rigid background and parts respectively. (c)-(d) An initial depth is refined by triangulating optical flow within each rigid motion mask. Note that the tower in the cyan rectangle is leaning in the initial MiDaS [12] depth, but "rectified" by our method. (e)-(f) Our method segments rigid objects more reliably than the prior two-frame rigidity estimation method [10] and generalizes to novel appearance compared to appearance-based detectors [8].

5. Qualitative comparison

We provide additional visual comparisons with prior approaches on KITTI and Sintel in Fig. 7 and Fig. 8. Compared to appearance-based methods for segmenting rigid motions, our method is able to correctly segment the static objects as part of the rigid background, and generalizes to novel appearance. Compared to geometric motion segmentation methods, our method is more robust to degenerate motion configurations and noisy flow as well as camera inputs. One example of our method decomposing a flying dragon into multiple rigid parts is shown in Fig. 2.

6. Ablation study of rigid body scene flow

We study the effect of rigid motion parameterization for scene flow estimation and report results on 200 images of KITTI-SF as shown in Tab. 2. Without rigid motion parameterization, our method is equivalent to optical expansion [18], which upgrades 2D flow fields to 3D, but does not refine the first frame disparity as well as optical flow. In contrast, the proposed method reduces the overall scene flow error by 24.6% through rigid body refinement. Replacing the proposed rigid motion masks with appearance-based masks pro-

Table 2: Ablation study of stereo scene flow on KITTI-SF images. D1 and D2: first and second frame disparity error. Fl: optical flow error. all: evaluated on all pixels. fg: evaluated on foreground pixels only. SF: scene flow error. Δ: percentage of error reduction after refinement. *First frame disparity does not change during refinement.

	*D1 (%)		D2 (%)		Fl (%)				SF (%)			
Method	*all* ↓	*fg* ↓	*all* ↓	*fg* ↓	*all* ↓	Δ-all ↑	*fg* ↓	Δ-fg ↑	*all* ↓	Δ-all ↑	*fg* ↓	Δ-fg ↑
Baseline OE [18]	1.41	0.76	2.45	**0.91**	4.02	0	2.50	0	5.12	0	3.07	0
Ours Mask R-CNN	1.41	0.76	2.11	1.99	3.53	12.1	4.34	-73.6	4.02	21.5	4.86	-36.8
Ours Rigid Mask	1.41	0.76	**2.04**	1.05	**3.32**	**17.4**	**2.16**	**15.7**	**3.86**	**24.6**	**2.78**	**10.4**

duced by Mask R-CNN leads to a noticeable accuracy drop. Our rigid body parameterization also leads to a constant improvement of scene flow accuracy for both foreground and background regions.

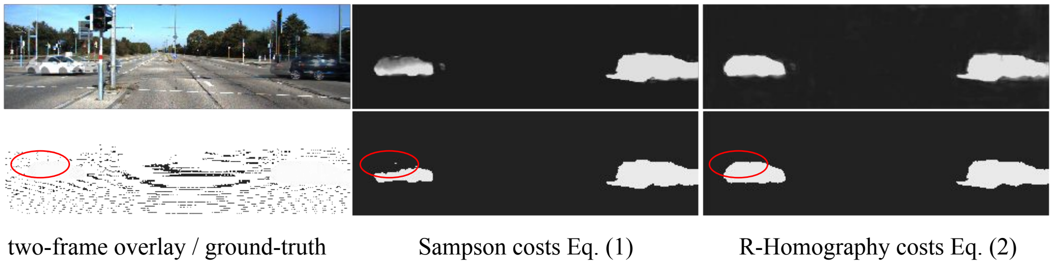


Figure 3: Epipolar costs vs homography costs. **Top**: Gray-scale costs values. **Bottom**: Binary segmentations after thresholding the costs. This scene features two moving foreground cars and a static camera that causes motion degeneracy in epipolar geometry (e.g., the low-cost but moving region in the Sampson cost map, marked by the red circle). In such cases, epipolar line is not well-defined, and the homography model is more suitable for motion segmentation.

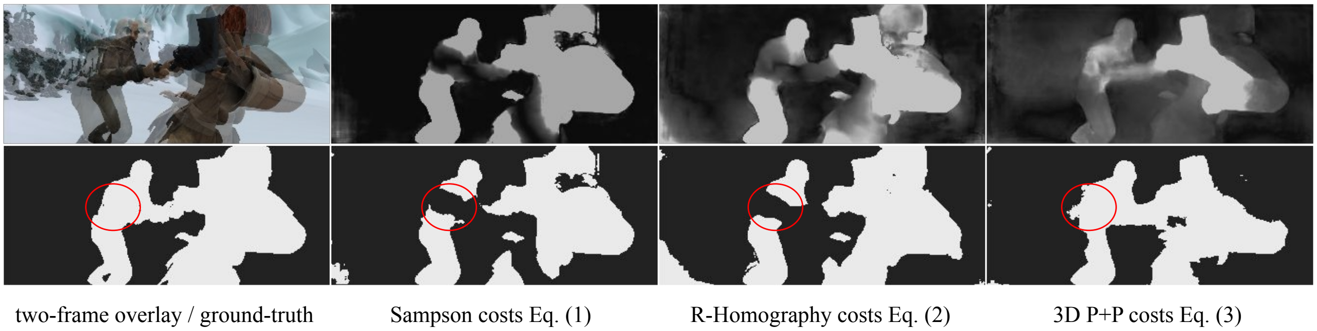


Figure 4: 3D P+P costs. **Top**: Gray-scale costs. **Bottom**: Binary segmentations after thresholding the costs. This scene contains a moving camera and nonrigid dynamic objects, where pixels that move along the epipolar line are not recoverable under classic motion segmentation criteria (e.g., the low-cost but moving region in the Sampson cost map, marked by the red circle). In such cases, our 3D P+P cost is more suitable.

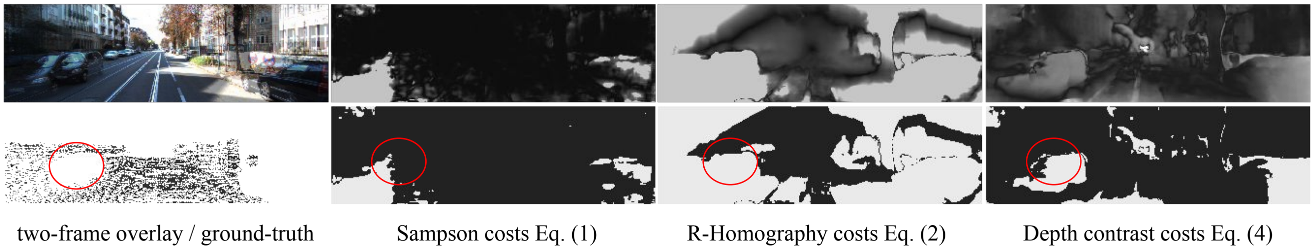

two-frame overlay / ground-truth
Sampson costs Eq. (1)
R-Homography costs Eq. (2)
Depth contrast costs Eq. (4)

Figure 5: Depth contrast costs. **Top**: Gray-scale costs. **Bottom**: Binary segmentations after thresholding the costs. This scene contains a moving camera and a rigid body (car) moving along the negative direction of camera translation, which is not recoverable under classic motion segmentation criteria (e.g., the low-cost but moving region in the Sampson cost map, marked by the red circle) as well as the 3D P+P cost. In such cases, our depth contrast cost is more suitable.

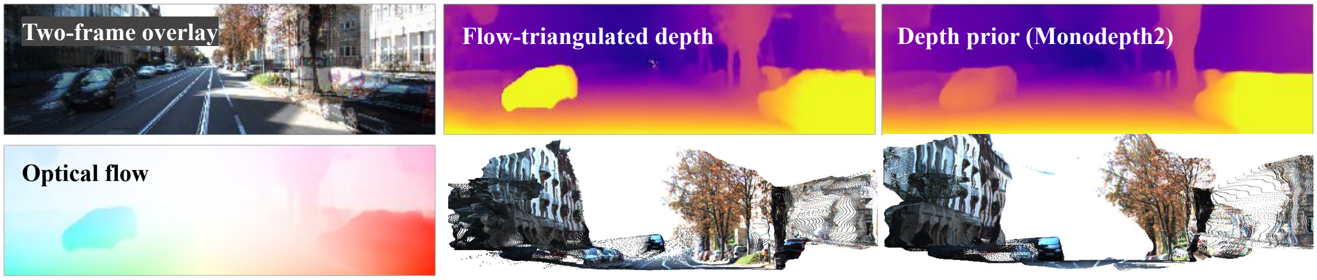


Figure 6: Flow-triangulated depth vs monocular depth prior. The flow-triangulated depth Z_0^{flow} (middle) is the triangulation of motion correspondences assuming overall rigidity. The monocular depth prior Z_0^{prior} (right) can be represented by a data-driven monocular depth network [12]. In this example, the left vehicle is moving opposite to the camera translation direction, and cannot be detected by epipolar constraints, as shown in Fig. 2 of the main text. However, it appears abnormal (floating above the ground) in the flow-triangulated reconstruction. To detect such collinearly moving objects, we globally align Z_0^{prior} to Z_0^{flow} by a scale factor γ computed as the ratio of their medians, which reveals the floating (moving) car that is inconsistent with the monocular depth prior.

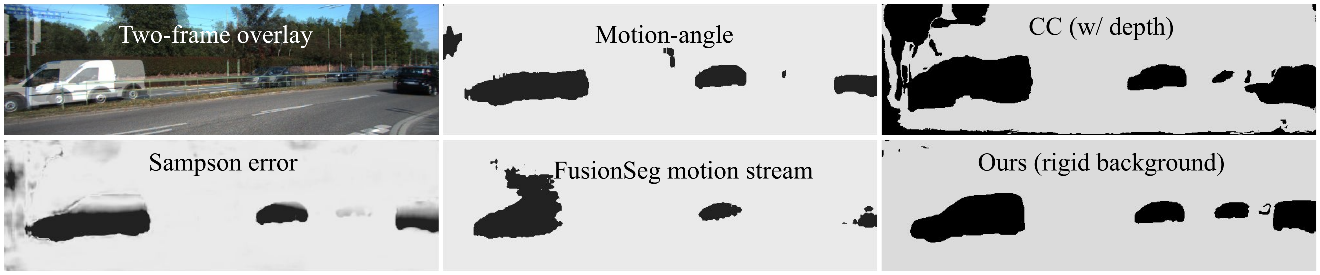


Figure 7: Illustration of coplanar motion ambiguity on KITTI. Rigid background are indicated by the white color. Points moving along the epipolar line, for example the roof of the cars, yields small Sampson error, and therefore are estimated as background in classic geometric pipelines. We make use of optical expansion, which reveals the relative depth change, to resolve such ambiguity. Compared to prior motion-based segmentation methods, ours is more robust to noise.

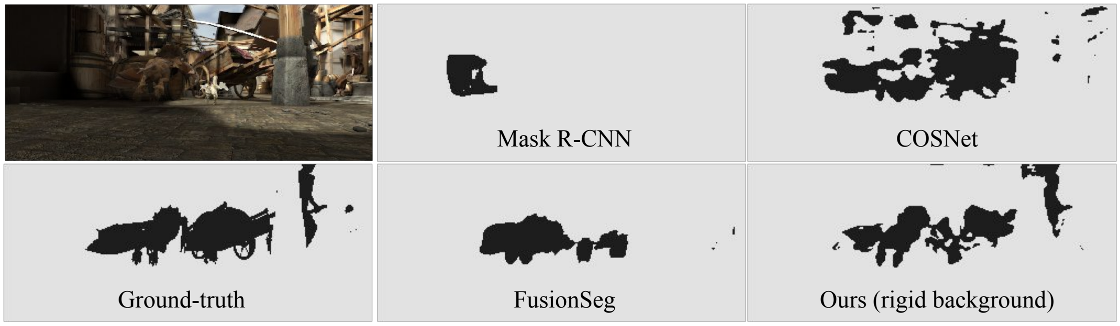


Figure 8: Results on Sintel market sequence. Prior single frame or video motion segmentation methods fail due to unusual view-point (viewing from the ground) and never-before-seen objects (dragon, wood carts). Our method accurately segments novel moving objects.

References

[1] Pia Bideau and Erik Learned-Miller. It's moving! a probabilistic model for causal motion segmentation in moving camera videos. In *ECCV*, 2016. 1

[2] Qi Cai, Yuanxin Wu, Lilian Zhang, and Peike Zhang. Equivalent constraints for two-view geometry: pose solution/pure rotation identification and 3d reconstruction. *IJCV*, 2019. 1

[3] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 1

[4] Elan Dubrofsky. Homography estimation. *Diplomová práce. Vancouver: Univerzita Britské Kolumbie*, 2009. 1

[5] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2

[6] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth prediction. In *ICCV*, 2019. 1

[7] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 1, 2

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 2

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[10] Zhaoyang Lv, Kihwan Kim, Alejandro Troccoli, Deqing Sun, James Rehg, and Jan Kautz. Learning rigidity in dynamic scenes with a moving camera for 3d motion field estimation. In *ECCV*, 2018. 2

[11] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016. 1

[12] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *arXiv:1907.01341*, 2019. 2, 5

[13] Stephan R Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In *ICCV*, pages 2213–2222, 2017. 1

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1

[15] Peter J Rousseeuw. Least median of squares regression. *Journal of the American statistical association*, 79(388):871–880, 1984. 2

[16] Philip HS Torr, Andrew W Fitzgibbon, and Andrew Zisserman. The problem of degeneracy in structure and motion recovery from uncalibrated image sequences. *IJCV*, 1999. 1

[17] Gengshan Yang and Deva Ramanan. Volumetric correspondence networks for optical flow. In *NeurIPS*, 2019. 1, 2

[18] Gengshan Yang and Deva Ramanan. Upgrading optical flow to 3d scene flow through optical expansion. In *CVPR*, 2020. 1, 2, 3

[19] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *CVPR*, pages 2403–2412, 2018. 1

[20] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv preprint arXiv:1904.07850*, 2019. 1