Instance-wise Occlusion and Depth Orders in Natural Scenes
Supplementary Material

Hyunmin Lee
LG AI Research
hyunmin@lgresearch.ai

Jaesik Park
POSTECH GSAI & CSE
jaesik.park@postech.ac.kr

A1. Evaluation Metrics

A1.1. Occlusion order recovery

We evaluate the occlusion order of every instance pair using recall, precision, and F1 score. In particular, we report the accuracy of predicting which of the two instances is an occluder, as done in OrderNet$^{\text{M+I}}$ [12] and PCNet-M [11].

Recall is computed as the number of correctly predicted occluding orders divided by the number of ground truth occluding orders. Precision is the number of correctly predicted occluding orders divided by the total number of predicted occluding orders. F1 score is the harmonic mean of precision and recall. The equation of *Recall*, *Precision* and *F1* score are defined as follows:

$$\text{Recall} = \frac{\sum_{AB} \mathbb{1}(o'_{AB} = 1 \text{ and } o_{AB} = 1)}{\sum_{AB} \mathbb{1}(o_{AB} = 1)}, \tag{A1}$$

$$\text{Precision} = \frac{\sum_{AB} \mathbb{1}(o'_{AB} = 1 \text{ and } o_{AB} = 1)}{\sum_{AB} \mathbb{1}(o'_{AB} = 1)}, \tag{A2}$$

$$\text{F1 score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{A3}$$

where o and o' denote ground truth and predicted occlusion order, and $\mathbb{1}$ is an indicator function.

A1.2. Depth order recovery

We evaluate depth order recovery accuracy using Weighted Human Disagreement Rate (WHDR) [1], which represents the percentage of weighted disagreement between ground truth d and predicted depth order d'. The weights are proportional to the confidence of each annotation. Here, we use the inverse of *count* multiplied by the minimum number of participants. WHDR evaluates {closer, equal, farther} relation on each of {distinct, overlap, or all} categories separately; which is defined as follows:

$$\begin{aligned} \text{WHDR} &= \frac{\sum_{AB} w_{AB} \cdot \mathbb{1}(d'_{AB} \neq d_{AB})}{\sum_{AB} w_{AB}}, \\ \text{where } w_{AB} &= \frac{2}{count_{AB}}. \end{aligned} \tag{A4}$$

A1.3. Disparity map prediction

For the disparity map prediction on the KITTI dataset [4], we evaluate the performance of our InstaDepthNet and MiDaS [9] using following metrics:

$$\text{Abs Rel} = \frac{1}{|\mathcal{G}|} \sum_{i \in \mathcal{G}} \frac{|D'(i) - D(i)|}{D(i)}, \tag{A5}$$

$$\text{Sq Rel} = \frac{1}{|\mathcal{G}|} \sum_{i \in \mathcal{G}} \frac{||D'(i) - D(i)||^2}{D(i)}, \tag{A6}$$

$$\text{RMSE log} = \sqrt{\frac{1}{|\mathcal{G}|} \sum_{i \in \mathcal{G}} ||\log D'(i) - \log D(i)||^2}, \tag{A7}$$

$$\begin{gathered} \text{Accuracy} = \% \text{ of } D'(i) \text{ s.t.} \\ \max\left(\frac{D'(i)}{D(i)}, \frac{D(i)}{D'(i)}\right) = \delta < \tau, \end{gathered} \tag{A8}$$

where D and D' denote ground truth and predicted depth maps, and $\mathcal{G}$ indicates the pixels whose ground truth values are available.

A2. Additional results

A2.1. Bidirectional occlusion order

We conduct experiments with the INSTAORDER dataset to verify the effect of bidirectional occlusion orders. We compare the accuracy with and without using the bidirectional occlusion orders for both training and testing (Table A1). Intuitively, classifying smaller occlusion order categories (no occlusion, A→B, B→A) seems more manageable, but methods not using bidirectional order reported lower scores than those using bidirectional order. We speculate that bidirectional order helps to distinguish ambiguous ordering cases.

Methods	Occ. order			Occ. acc. ↑			Depth WHDR ↓		
	No	Uni	Bi	Recall	Prec.	F1	Distinct	Overlap	All
PCNet-M [11]	✓	✓		62.23	57.74	52.28	-	-	-
OrderNet^{M+I} [12]	✓	✓		88.68	62.90	66.23	-	-	-
InstaOrderNet^{o}	✓	✓		89.23	67.08	69.34	-	-	-
InstaDepthNet^{o,d}	✓	✓		79.76	89.39	78.13	7.09	23.46	12.79
PCNet-M [11]	✓	✓	✓	59.19	76.42	63.02	-	-	-
OrderNet^{M+I} (ext.)	✓	✓	✓	84.93	78.21	77.51	-	-	-
InstaOrderNet^{o}	✓	✓	✓	89.39	79.83	80.65	-	-	-
InstaDepthNet^{o,d}	✓	✓	✓	**84.89**	**91.34**	**85.01**	**7.00**	**23.29**	**12.72**

Table A1. Ablation study of utilizing bidirectional occlusion orders for occlusion and depth order prediction tasks.

Loss weights			InstaOrder			DIW		
$\mathcal{L}_{do}$	$\mathcal{L}_{disp}$	$\mathcal{L}_{s}$	WHDR Distinct ↓	WHDR Overlap ↓	WHDR All ↓	Correct ↑	Wrong ↓	WHDR ↓
1	0	0.1	7.24	23.99	13.17	65,270	9,171	12.32
1	1	0	**7.14**	23.64	13.00	65,277	9,164	12.30
1	1	0.1	7.25	**23.34**	**12.94**	**65,317**	**9,124**	**12.26**

Table A2. Ablation study on losses applied to InstaDepthNet^{d}.

A2.2. Loss functions

We conduct an ablation study on loss functions to validate the effectiveness of our proposed instance-wise disparity loss. We train InstaDepthNet^{d} with varying losses using InstaOrder training set. Then we report WHDR using InstaOrder validation set and DIW test set. L_{do} is depth order loss, L_{disp} is the proposed instance-wise disparity loss, and L_s is edge-aware smoothness loss (Sec 4.2 in the main paper). Experimental result (Table A2) shows that accuracy degraded without L_{disp} or L_s. Especially, the absence of L_{disp} degraded the accuracy by a large margin, which demonstrates the usefulness of the proposed instance-wise disparity loss.

A3. InstaOrder Information

A3.1. License

We constructed the InstaOrder dataset utilizing COCO 2017 [7] images and instance masks. COCO 2017 annotations are licensed under a CC BY 4.0 license. Image source of COCO 2017 is Flickr, and the copyrights follow Flickr's terms of use[1]. Similarly, our annotations in InstaOrder are licensed under a CC BY 4.0 license.

A3.2. Guideline

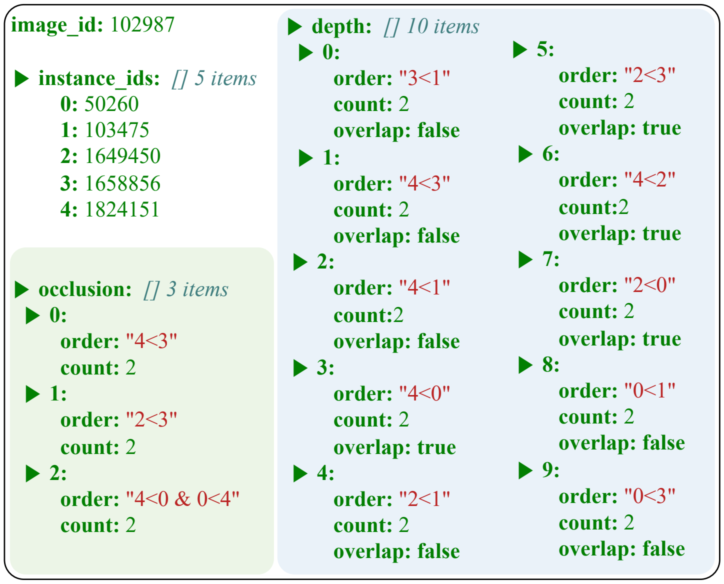
We provide a guideline to annotators: we ask them to annotate only semantically meaningful instances and to consider the entire structure of instances. Some instance pairs are unclear to annotate occlusion and depth order. For example, (i) a collage image (multiple photos appear in one photo) and (ii) objects shown on television, magazine, or mirror. Images of case (i) are discarded, and instances in case (ii) are annotated as equally distant without occlusion. To eliminate the bias from the image sequence, we provide randomly shuffled images for each annotator.

A3.3. Wages

We annotated InstaOrder by crowd-sourcing, and the total amount of money given to workers is $35,000. The workers are paid based on the number of annotations. Before the crowd-sourcing begins, we monitor unprofessional workers and measure the average time taken for one annotation to set the proper reward. We set different rewards for occlusion and depth order annotation based on the time.

After crowd-sourcing finishes, we check the actual annotation times by crowd workers. On average, it took 2.68 seconds for a single occlusion order annotation and 5.05 seconds for a single depth order annotation. With this speed, the hourly rewards we give are $6 for the occlusion order task and $4.5 for the depth order task. We also provide $45 for each of the top 50 depth order annotators to promote the task. This reward design motivated crowd workers, and our task was popular on the crowd-sourcing platform. As a result, a total of 3,659 workers participated in the task, and the annotation job just took a month.

[1] https://www.flickr.com/creativecommons/



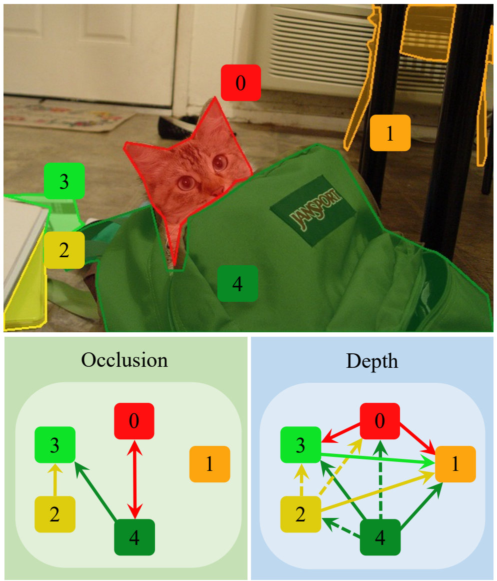
```
image_id: 102987

▶ instance_ids: [] 5 items
    0: 50260
    1: 103475
    2: 1649450
    3: 1658856
    4: 1824151

▶ occlusion: [] 3 items
  ▶ 0:
      order: "4<3"
      count: 2
  ▶ 1:
      order: "2<3"
      count: 2
  ▶ 2:
      order: "4<0 & 0<4"
      count: 2

▶ depth: [] 10 items
  ▶ 0:
      order: "3<1"
      count: 2
      overlap: false
  ▶ 1:
      order: "4<3"
      count: 2
      overlap: false
  ▶ 2:
      order: "4<1"
      count:2
      overlap: false
  ▶ 3:
      order: "4<0"
      count: 2
      overlap: true
  ▶ 4:
      order: "2<1"
      count: 2
      overlap: false
  ▶ 5:
      order: "2<3"
      count: 2
      overlap: true
  ▶ 6:
      order: "4<2"
      count:2
      overlap: true
  ▶ 7:
      order: "2<0"
      count: 2
      overlap: true
  ▶ 8:
      order: "0<1"
      count: 2
      overlap: false
  ▶ 9:
      order: "0<3"
      count: 2
      overlap: false
```

(a) json file provided by INSTAORDER dataset



(b) Instance-wise occlusion and depth orders

Figure A1. Overview of the proposed INSTAORDER dataset.

A3.4. Data example

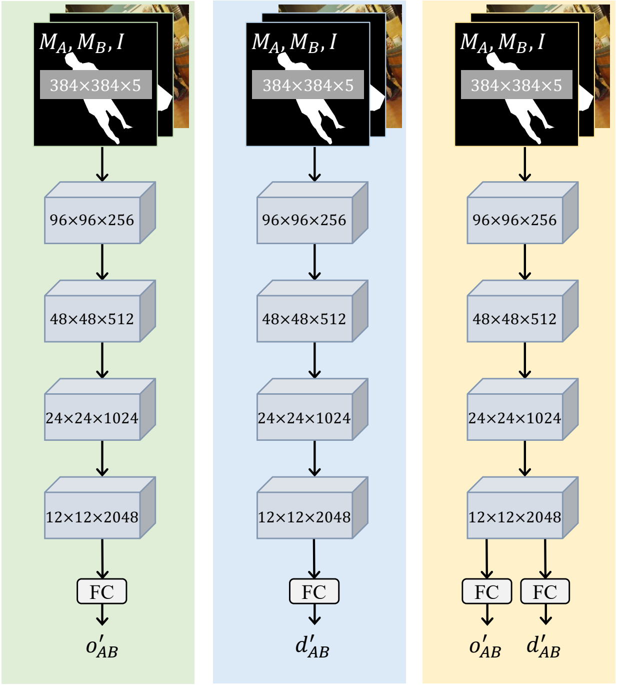
As noted, INSTAORDER is annotated upon COCO 2017 [7] dataset, and therefore only the json file is provided (Figure A1, a). For an image (image_id), five instances (instance_ids) with class labels are from the COCO dataset. With this information, we denote the occlusion order as ”occluder_id < occludee_id”, and for bidirectional order ”A<B & B<A” notation is used. Similarly, depth order is denoted as ”closer_id < farther_id” and for equal depth ”A=B” notation is used. Besides the orders, we also provide the metadata such as count and overlap. As a result, we can generate occlusion and depth graphs (Figure A1, b).

A4. Implementation Details

A4.1. Training details

We train InstaOrderNet with SGD optimizer [2] for 58K iterations. The initial learning rate set to 0.001 is decayed by 0.1 after 32K and 48K iterations. InstaOrderNet use ResNet-50 [6] initialized with Xavier init [5]. We use a batch size of 128 distributed over four Nvidia TITAN RTX GPUs.

InstaDepthNet consists of two heads: the order prediction head and the depth map prediction head. The order prediction head uses ResNet-50 [6] and the depth map prediction head uses MiDaS-v2 [9]. Similar to InstaOrderNet, ResNet-50 [6] is utilized with Xavier init [5]. For MiDaS-v2 [9], we adopt pre-trained weights[2] provided by the authors. We train InstaDepthNet with a batch size of 48 using four Nvidia V100 GPUs. The initial learning rate is set to 0.0001.



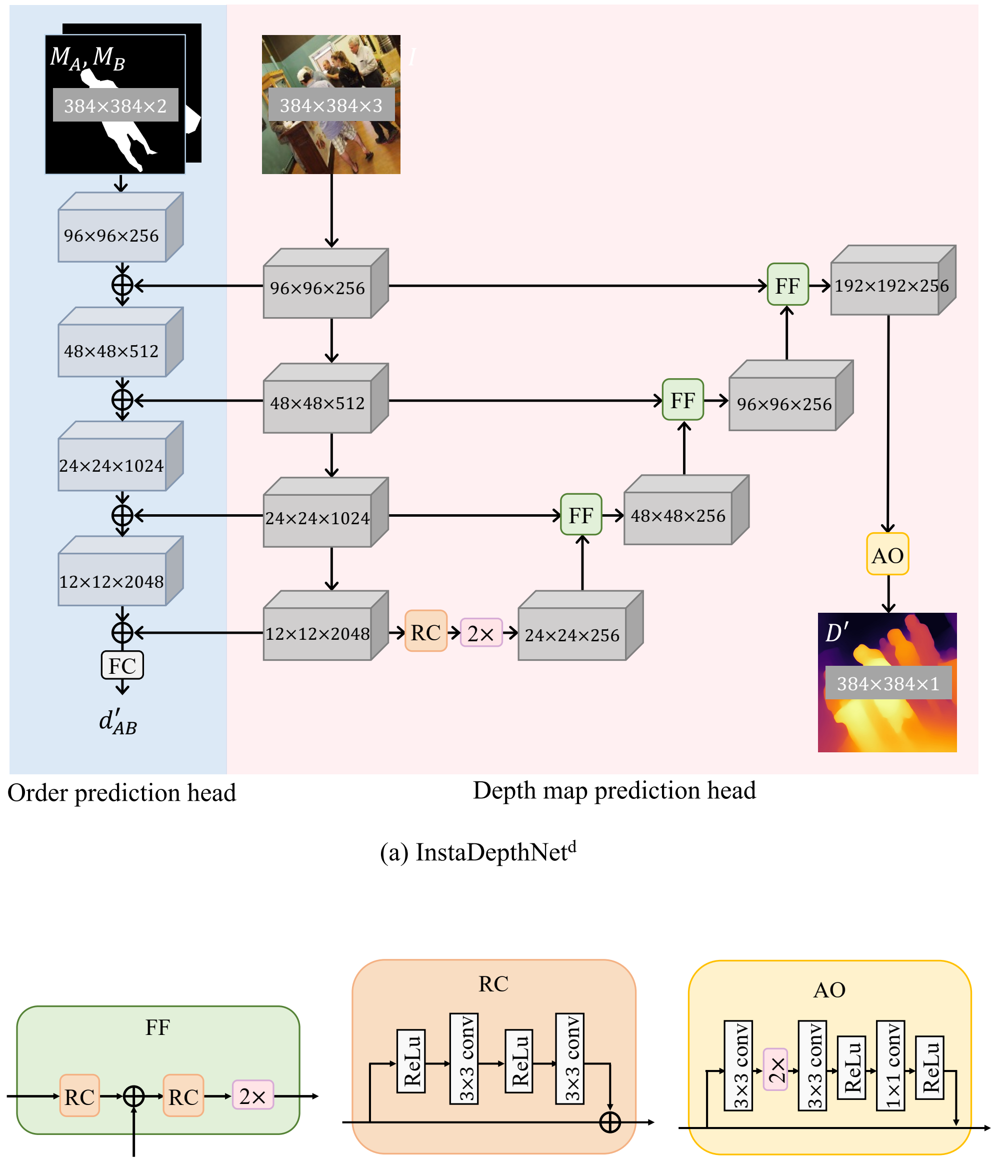
(a) InstaOrderNet^o (b) InstaOrderNet^d (c) InstaOrderNet^{o,d}

Figure A2. InstaOrderNet model architecture.

A4.2. InstaOrderNet architecture

InstaOrderNet (Figure A2) takes pairwise instance masks (M_A, M_B) and an image (I) as input and outputs their instance-wise orders. InstaOrderNet uses ResNet-50 [6] as noted. Here we denote the feature size by [height, width, channel].

[2] https://github.com/intel-isl/MiDaS



(a) InstaDepthNet$^{\mathrm{d}}$

(b) Feature fusion

(c) Residual convolution

(d) Adaptive output

Figure A3. InstaDepthNet$^{\mathrm{d}}$ model architecture.

A4.3. InstaDepthNet architecture

InstaDepthNet$^{\mathrm{d}}$ (Figure A3) takes pairwise instance masks (M_A, M_B) and an image (I) as input and outputs their depth order($d^{'}_{AB}$) along with a disparity map $(D^{'})$. The depth order prediction head uses ResNet-50 [6], and the disparity map prediction head is MiDaS [9]. Please refer to the paper by Xian *et al*. [10] for a detailed explanation of MiDaS architecture.

InstaDepthNet$^{\mathrm{d}}$ architecture is modular because the order prediction head can be used optionally depending on the requirements at test time. Specifically, InstaDepthNet$^{\mathrm{d}}$ can produce a dense disparity map even when instance masks are unavailable, such as DIW [3] dataset.

A4.4. Input resolution.

For the networks that produce depth order (InstaOrder$^{\mathrm{d}}$, InstaOrder$^{\mathrm{o,d}}$, InstaDepthNet$^{\mathrm{d}}$ and InstaDepthNet$^{\mathrm{o,d}}$), we set image resolution as 384×384 by following the MiDaS [9]. On the other hand, for the network that does not produce depth order (InstaOrderNet$^{\mathrm{o}}$), we set the input size as described in PCNet-M [11]. Inputs of InstaOrderNet$^{\mathrm{o}}$ are patches that are adaptively cropped to contain objects at the center, then resized to 256×256 at the train and test time.

A5. License of Other Assets

COCO 2017 [7] annotations are licensed under a CC BY 4.0 license. Image source of COCO 2017 is Flickr, and the copyrights follow Flickr's terms of use[3]. We conducted experiments using the dataset COCOA [12], KINS [8], and DIW [3]. To our best knowledge, COCOA and KINS are publicly released as written in the papers [8, 12]. However, we could not find the appropriate license for the DIW [3] dataset.

We utilized a pre-trained model of MiDaS-v2 [9][4] that follows MIT license, and PCNet-M [11][5] that follows Apache License 2.0.

References

[1] Sean Bell, Kavita Bala, and Noah Snavely. Intrinsic images in the wild. *ACM Transactions on Graphics (SIGGRAPH)*, 2014. 1

[2] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *COMPSTAT*, 2010. 3

[3] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-Image Depth Perception in the Wild. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 4, 5

[4] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 1

[5] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, (AISTATS)*, 2010. 3

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3, 4

[7] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014. 2, 3, 5

[8] Lu Qi, Li Jiang, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Amodal Instance Segmentation With KINS Dataset. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5

[9] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 1, 3, 4, 5

[10] Ke Xian, Chunhua Shen, Zhiguo Cao, Hao Lu, Yang Xiao, Ruibo Li, and Zhenbo Luo. Monocular Relative Depth Perception With Web Stereo Data Supervision. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 4

[11] Xiaohang Zhan, Xingang Pan, Bo Dai, Ziwei Liu, Dahua Lin, and Chen Change Loy. Self-Supervised Scene De-Occlusion. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 4, 5

[12] Yan Zhu, Yuandong Tian, Dimitris N. Metaxas, and Piotr Dollár. Semantic Amodal Segmentation. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 5

[3] https://www.flickr.com/creativecommons/

[4] https://github.com/intel-isl/MiDaS

[5] https://github.com/XiaohangZhan/deocclusion/