Pixel-Pair Occlusion Relationship Map (P2ORM): Formulation, Inference & Application Supplementary Material

Xuchong Qiu[1], Yang Xiao[1], Chaohui Wang[1]*, Renaud Marlet[1,2]

[1]LIGM, Ecole des Ponts, Univ Gustave Eiffel, ESIEE Paris, CNRS, France
[2]valeo.ai, Paris, France *Corresponding author: `chaohui.wang@univ-eiffel.fr`

In this supplementary material, we provide:

A.1. ROC-like curves showing our significant and systematic outperformance;
A.2. The detailed architecture of our network for occlusion relation estimation;
A.3. An ablation study on domain adaptation for training with synthetic images;
A.4. An ablation study about the impact of ground truth quality
A.5. An adaption of ground truth generation with noised geometry information
B.1. Detailed quantitative results on the application to depth map refinement;
B.2. An ablation study regarding variants for depth map refinement;
C. Illustration of occlusion relationship annotations in our generated datasets.

A Pixel-pair occlusion relationship estimation

A.1 ROC-like comparison to the state of the art

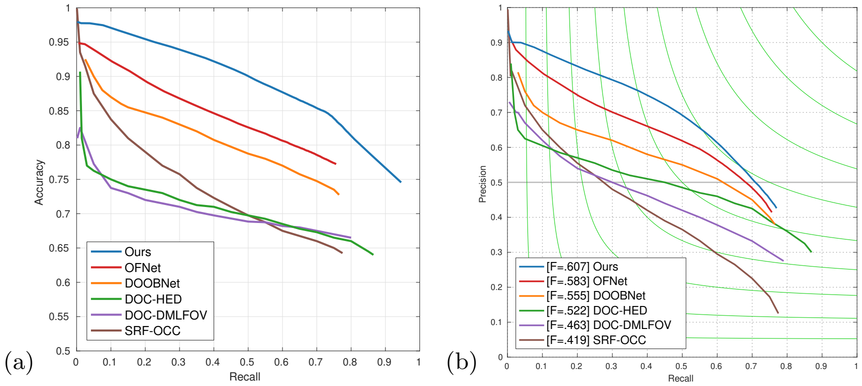

Fig. 9. Oriented occlusion boundary estimation on BSDS ownership: (a) Occlusion-Accuracy-Recall curve (AOR) [23], (b) Occlusion-Precision-Recall curve (OPR) [22].

To allow a deeper assessment of the performance of our approach, compared to other state-of-the-art methods, we plot two graphs (cf. Fig. 9):

(a) the Occlusion Accuracy w.r.t. boundary Recall (AOR) curve, as introduced in [23], represents accuracy as a function of recall;

(b) the Occlusion Precision w.r.t. boundary Recall (OPR) curve, as later proposed in [22], represents precision as a function of recall — a harder metric;

where:

- (R)ecall is the proportion of pixels with correct boundary detections;
- (P)recision is the proportion of pixels with correct occlusion orientation w.r.t. all pixels detected as occlusion boundary;
- (A)ccuracy is the proportion of pixels with correct occlusion orientation w.r.t. all pixels *correctly* detected as occlusion boundary.

We compared all methods using the BSDS ownership dataset. This dataset has become a de facto standard regarding oriented occlusion boundary estimation, despite its moderate size and its coarse manually-annotated ground truth. We use exactly the same dataset (same training data and same test data) for all methods, including ours, which is not trained here on the large InteriorNet-OR dataset that we generated. On both AOR and OPR curves, we largely outperform all other existing methods, i.e., SRF-OCC [20], DOC-DMLFOV [23], DOC-HED [23], DOOBNet [22] and OFNet [15].

A.2 Detailed architecture of our network for occlusion estimation

On Fig. 4 of the paper and in the text, we sketched the architecture of our network for pixel-pair occlusion relationship estimation (P2ORNet). Here we provide additional information. The detailed architecture is depicted on Fig. 10, while the setup of each block is presented in Tab. 3. Blocks named as "Res" are residual convolution blocks introduced in [6] and blocks named as "Deconv" are transposed convolution layers.

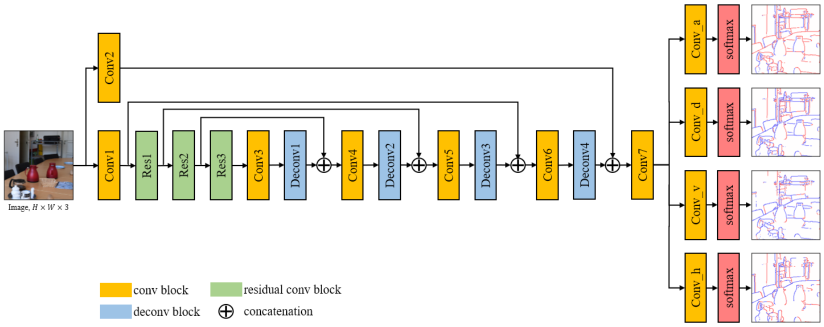

Fig. 10. Architecture of P2ORNet, for occlusion relationship estimation.

We initialize the encoder model of the occlusion estimation module with the weights of a ResNet-50 model [6] pre-trained on ImageNet, and the remaining layers with random values (as defined by the PyTorch default initialization). To

Conv1	Conv2	Res1	Res2	Res3	Conv3
$[7{\times}7, 64] \times 1$ stride 2	$\begin{bmatrix} 3{\times}3, 64 \\ 3{\times}3, 64 \end{bmatrix} \times 1$	$3{\times}3$ maxpool, stride 2 $\begin{bmatrix} 1{\times}1, 64 \\ 3{\times}3, 64 \\ 1{\times}1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1{\times}1, 128 \\ 3{\times}3, 128 \\ 1{\times}1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1{\times}1, 256 \\ 3{\times}3, 256 \\ 1{\times}1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 3{\times}3, 512 \\ 3{\times}3, 512 \end{bmatrix} \times 1$
Deconv1	Conv4	Deconv2	Conv5	Deconv3	Conv6
$[3{\times}3, 512] \times 1$ stride 2	$\begin{bmatrix} 3{\times}3, 256 \\ 3{\times}3, 256 \end{bmatrix} \times 1$	$[3{\times}3, 256] \times 1$ stride 2	$\begin{bmatrix} 3{\times}3, 64 \\ 3{\times}3, 64 \end{bmatrix} \times 1$	$[3{\times}3, 64] \times 1$ stride 2	$\begin{bmatrix} 3{\times}3, 64 \\ 3{\times}3, 64 \end{bmatrix} \times 1$
Deconv4	Conv7	Conv_h	Conv_v	Conv_d	Conv_a
$[3{\times}3, 64] \times 1$ stride 2	$\begin{bmatrix} 3{\times}3, 64 \\ 3{\times}3, 64 \end{bmatrix} \times 1$	$[1{\times}1, 3] \times 1$	$[1{\times}1, 3] \times 1$	$[1{\times}1, 3] \times 1$	$[1{\times}1, 3] \times 1$

Table 3. Detailed architecture of P2ORNet (pixel-pair occlusion relation estimation). For each block, as in [6], we give the kernel size and the number of output channels. The blocks Conv1, Res1, Res2, Res3 are the first four blocks of ResNet-50 [6].

NYUv2 (adapt. & test)	w/o adaptation			with adaptation			gain		
Method \ Metric	ODS	OIS	AP	ODS	OIS	AP	ODS	OIS	AP
DOOBNet*	.292	.324	.204	.343	.370	.263	.051	.046	.059
OFNet*	.339	.366	.255	.402	.431	.342	.063	.065	.087
baseline	.394	.418	.336	.396	.428	.343	.002	.010	.007
ours (4-connectivity)	.425	.446	.369	.500	.522	.477	.075	.076	.108
ours (8-connectivity)	.452	.477	.424	.520	.540	.497	.068	.063	.073

Table 4. Ablation study on domain adaptation using [26] with NYUv2 [16] images as target for training on synthetic images of InteriorNet [12] and testing on NYUv2. *Our re-implementation (cf. footnote in Sect. 5 of the paper). In blue, the minimum gain; in red, the maximum gain.

train the network, we use the ADAM optimizer [8] with learning rate 10^{-4} and divide it by 10 when half of the total training iterations (4000, 100000, 110000 for BSDS, NYUv2-OR, iBims-1-OR respectively) is reached. The input image size during training is 320×320, and the mini-batch size is 8.

A.3 Ablation study on domain adaptation for synthetic images

To evaluate on NYUv2 [16] and iBims-1 [9], we train on 10^4 synthetic images of InteriorNet [12] (cf. Sect. 5 and Tab. 1 of the paper). The pictures in InteriorNet are not totally photorealistic, but still fairly good. In our experiments on iBims-1, whose test images are of good quality, we train directly on InteriorNet images and get good results. However, on NYUv2, the test images are of low quality, with some amount of blur. To get better results, we adapted the InteriorNet images using the training images of NYUv2 as target domain, using [26].

The quantitative results in Tab. 4 show that this domain adaptation is worthwhile: except for the "baseline" method, for which the gains are limited, we gain on all other methods at least 4.6 points and up to 10.8 points, depending on the considered metric.

Method	NYUv2-OR		
Metric	ODS	OIS	AP
ours (w/o order-1)	.404	.439	.368
ours	**.520**	**.540**	**.497**

Table 5. Ablation study on P2ORM ground truth generation.

A.4 Ablation study on ground truth generation

To illustrate the value of defining occlusion at order-1 as introduced in Sect. 5 of the paper, here we show quantitative results where the occlusion ground truths are generated by occlusion at order-0 definition. We train on 10^4 domain-adapted images of InteriorNet [12] (cf. Sect. 5 and Tab. 1 of the paper) and evaluate on NYUv2-OR dataset. As shown in Tab. 5, if the ground truths are generated without considering occlusion at order-1, the performance of the model degrades greatly due to inaccurate supervision signals.

A.5 Ground truth generation from depth acquired by laser scanner

In real datasets such as iBims-1 [9] whose depths are captured by laser scanner, the depth estimation noise varies with both actual depth, pixel spatial location and surface orientation due to laser scanner hardware properties. Therefore we propose the following formula (cf. Eq. 1) to calculate possible depth estimation error E_p relevant to pixel p considering aforementioned factors.

$$E_p = \frac{\eta}{\tan(\gamma)} d_p \tag{1}$$

where η is a constant representing laser scanner estimation angular noise, γ is the angle between ray L_p and tangent plane $\mathit{\Pi}_p$, d_p is the Euclidean distance between surface point X_p and camera center C estimated by laser scanner. Then the discontinuity threshold δ as introduced in Sect. 2 of the paper between a pixel-pair p, q can be calculated with E_p, E_q and a constant C_δ that ensures a minimum discontinuity (cf. Eq. 2).

$$\delta = E_p + E_q + C_\delta \tag{2}$$

As shown in Fig. 5 of the paper, by using proposed occlusion definition and dynamic discontinuity threshold between each pixel-pair, the generated occlusion ground truths are accurate in the scene with a large depth range. Specifically, for iBims-1 dataset, $\eta = 0.005\, rad$ in Eq. 1 and $C_\delta = 25\, mm$ in Eq. 2.

B Depth map refinement

Implementation details. We initialize our network layers using the 'kaiming' initialization as in [6] and trained the network from scratch. The training set

Depth estim. method	Refinement method	Boundaries(↓)			Depth Error(↓)			Depth Accuracy(↑)		
		ϵ_{acc}	ϵ_{comp}	rel	$\log_{10}$	RMS_{lin}	RMS_{log}	σ_1	σ_2	σ_3
Eigen *et al.* [3]	—	9.926	9.993	0.236	0.095	0.765	0.265	0.611	0.887	0.971
	DispField [17]	2.168	8.173	0.232	0.094	0.758	0.263	0.615	0.889	0.971
	ours	**1.715**	**6.048**	0.231	0.095	0.761	0.264	0.615	0.888	0.970
Laina *et al.* [10]	—	4.702	8.982	0.142	0.059	0.510	0.181	0.818	0.955	0.988
	DispField [17]	2.372	7.041	0.140	0.059	0.509	0.180	0.819	0.956	0.989
	ours	**1.976**	**6.423**	0.142	0.059	0.508	0.181	0.818	0.955	0.988
Fu *et al.* [4]	—	3.872	8.117	0.131	0.053	0.493	0.174	0.848	0.956	0.984
	DispField [17]	3.001	7.242	0.136	0.054	0.502	0.178	0.844	0.954	0.983
	ours	**2.631**	**6.507**	0.132	0.053	0.487	0.173	0.848	0.957	0.985
Ramamonjisoa and Lepetit [18]	—	3.041	8.692	0.116	0.053	0.448	0.163	0.853	0.970	0.993
	DispField [17]	1.838	6.730	0.117	0.054	0.457	0.165	0.848	0.970	0.993
	ours	**1.546**	**5.988**	0.116	0.053	0.448	0.163	0.852	0.970	0.993
Jiao *et al.* [7]	—	8.730	9.864	0.093	0.043	0.356	0.134	0.908	0.981	0.995
	DispField [17]	2.410	8.230	0.092	0.042	0.352	0.132	0.910	0.981	0.995
	ours	**1.985**	**6.990**	0.093	0.042	0.351	0.133	0.909	0.981	0.995
Yin *et al.* [25]	—	1.854	7.188	0.112	0.047	0.417	0.144	0.880	0.975	0.994
	DispField [17]	1.762	6.307	0.112	0.047	0.419	0.144	0.879	0.975	0.994
	ours	**1.544**	**5.453**	0.113	0.047	0.421	0.145	0.878	0.975	0.994

Table 6. Evaluation of depth refinement on the output of several state-of-the-art methods on NYUv2 [16], cropped within valid region as in [3]. Best results in **bold**.

contains 10k depth predictions of SharpNet [18] on the InteriorNet subset [12] we consider, and corresponding ground-truth pixel-pair occlusion relationships in InteriorNet-OR. We used the ADAM optimizer [8] with a fixed learning rate of 10^{-5} and stopped training after 40k iterations. The size of the input depth images size is 640 × 480 and the batch size is 8 for all experiments.

B.1 Quantitative and qualitative results for depth refinement

Evaluation on NYUv2. Quantitative results of depth refinement on NYUv2 are shown in Tab. 6 for all metrics and for input depth maps obtained from a wide range of state-of-the-art depth estimation methods. After refinement, as mentioned in the paper (Sect. 5), the improvement or degradation of general accuracy metrics (i.e., "Depth Error" and "Depth Accuracy") are negligible (≤ 0.006 difference). This result is similar to the other depth refinement method, namely DispField [17]. However, we significantly and systematically outperform DispField on "Boundaries" metrics for the whole range of depth estimation methods. To further validate the effectiveness of P2ORM as depth refinement guidance, we also compare many existing methods using image intensity as guidance [21, 5, 1, 24, 19] where the initial depth prediction is given by [3]. Quantitative results of depth refinement on NYUv2 are shown in Tab. 7, our method achieves the best refinement results on all metrics.

Fig. 8 of the paper displays examples of refinements on iBims-1, with depth maps from SharpNet [18] as input, which is the second best method regarding boundary metrics ϵ_{acc} and ϵ_{comp}. We illustrate here, on Fig. 11, examples of refinements on NYUv2.

Method	Boundaries(↓)		Depth Error(↓)				Depth Accuracy(↑)		
	ϵ_{acc}	ϵ_{comp}	rel	$\log_{10}$	RMS_{lin}	RMS_{log}	σ_1	σ_2	σ_3
Initial estimation [3]	9.926	9.993	0.236	0.095	0.765	0.265	0.611	0.887	**0.971**
Bilateral Filter [21]	9.313	9.940	0.236	0.095	0.765	0.265	0.611	0.887	**0.971**
GF [5]	6.106	9.617	0.237	0.095	0.767	0.265	0.610	0.885	**0.971**
FBS [1]	5.428	9.454	0.236	0.095	0.765	0.264	0.611	0.887	**0.971**
Deep GF [24]	4.318	9.597	0.306	0.116	0.917	0.362	0.508	0.823	0.948
PACNet [19]	4.681	9.702	0.238	0.096	0.771	0.267	0.608	0.885	**0.971**
Ours	**1.715**	**6.048**	**0.231**	**0.095**	**0.761**	**0.264**	**0.615**	**0.888**	**0.971**

Table 7. Comparison with existing methods for image enhancement, adapted to the depth map refinement problems on NYUv2 [16] where initial depth estimation is given by [3]. Best results in **bold**.

Depth estimation method	Refinement method	Boundaries(↓)		Depth Error(↓)			Depth Accuracy(↑)		
		ϵ_{acc}	ϵ_{comp}	rel	$\log_{10}$	RMS_{lin}	σ_1	σ_2	σ_3
	—	9.97	9.99	0.32	0.17	1.55	0.36	0.65	0.84
Eigen *et al.* [3]	DispField	4.83	8.78	0.32	0.17	1.54	0.37	0.66	0.85
	ours	**2.46**	**5.74**	0.32	0.17	1.55	0.36	0.65	0.84
	—	6.19	9.17	0.26	0.13	1.20	0.50	0.78	0.91
Laina *et al.* [10]	DispField	3.32	7.15	0.25	0.13	1.20	0.51	0.79	0.91
	ours	**2.56**	**6.20**	0.26	0.13	1.20	0.50	0.78	0.90
	—	2.42	7.11	0.30	0.13	1.26	0.48	0.78	0.91
Liu *et al.* [14]	DispField	**2.36**	7.00	0.30	0.13	1.26	0.48	0.77	0.91
	ours	2.37	**5.91**	0.30	0.13	1.26	0.48	0.78	0.91
	—	3.90	8.17	0.22	0.11	1.09	0.58	0.85	0.94
Li *et al.* [11]	DispField	3.43	7.19	0.22	0.11	1.10	0.58	0.84	0.94
	ours	**2.07**	**5.26**	0.22	0.11	1.10	0.58	0.84	0.94
	—	4.84	8.86	0.29	0.17	1.45	0.41	0.70	0.86
Liu *et al.* [13]	DispField	2.78	7.65	0.29	0.17	1.47	0.40	0.69	0.86
	ours	**2.75**	**6.40**	0.29	0.17	1.45	0.41	0.69	0.86
Ramamonjisoa	—	3.69	7.82	0.27	0.11	1.08	0.59	0.83	0.93
and	DispField	**2.13**	6.33	0.27	0.11	1.08	0.59	0.83	0.93
Lepetit [18]	ours	2.16	**5.82**	0.27	0.11	1.08	0.59	0.83	0.93

Table 8. Evaluation of depth refinement on the output of several state-of-the-art methods on iBims-1 [9], cropped within valid region as in [3]. Best results in **bold**.

Evaluation on iBims-1. We evaluate and compare our method to DispField [17] on iBims-1, in the same setting as for NYUv2 above, cf. Tab. 8. The results are similar: after refinement, the improvement or degradation of general accuracy metrics are negligible (≤ 0.02 difference); however, we significantly and almost systematically outperform DispField [17] on “Boundaries” metrics for the whole range of depth estimation methods.

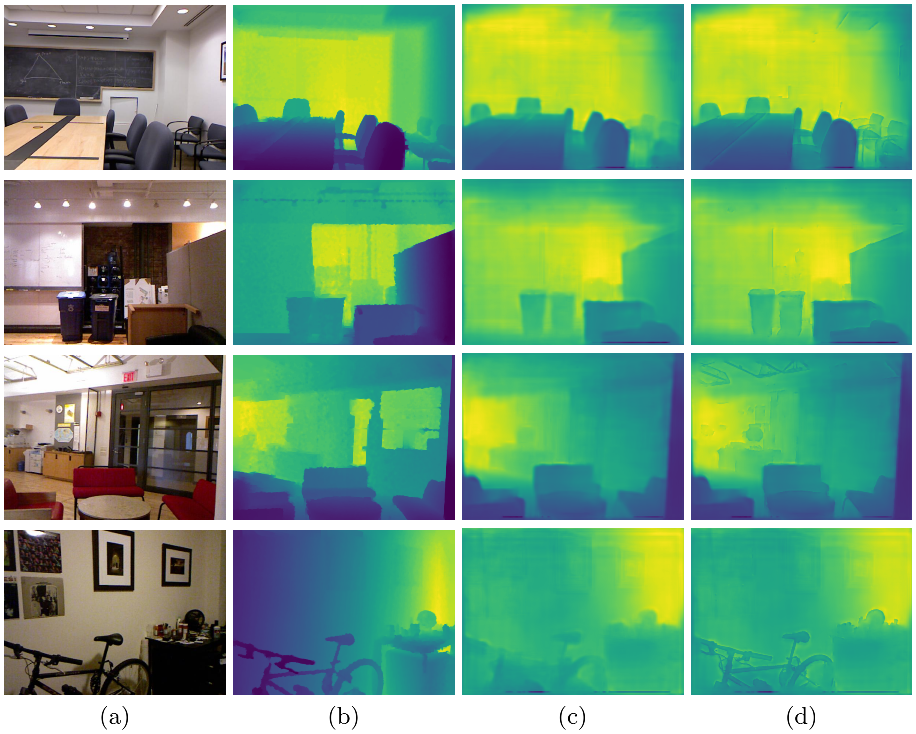

Fig. 11. Depth refinement on dataset NYUv2: (a) input RGB image from NYUv2, (b) ground truth depth, (c) SharpNet depth estimation [18], (d) our refined depth.

B.2 Alternative designs and ablation study for depth refinement

Many variants and alternatives are possible to exploit our pixel-pair occlusion relationships for depth map refinement. We report here quantitative results justifying the particular choice we made in Section 4 of the paper.

To evaluate the effectiveness of different variants, we consider as input the estimation of [7] as this method provides the depth maps with the best accuracy on the NYUv2 dataset. But the conclusion is still valid for other methods.

Alternative network inputs. We first explore the influence of other types of input, in place of our pixel-pair occlusion relationships: the original RGB image, a normal map estimated using [2], and a classical occlusion edge mask (i.e., a binary map). The occlusion edge masks are created by thresholding the occlusion boundaries derived from the estimated occlusion relationships after Non Maximal Suppression (NMS), as described in the paper. The network architecture and loss function are unchanged w.r.t. our proposed method, except that the first convolutional layer is adapted according to the number of input channels (1 more for the RGB image and the edge map, 3 more for the normal map).

As shown in the top part of Tab. 9, using the RGB image as input (line "RGB") instead of our pixel-pair occlusion relationships (line "Refined (ours)")

Variant	Boundaries(↓)		Depth error(↓)				Depth accuracy(↑)		
	ϵ_{acc}	ϵ_{comp}	rel	$\log_{10}$	RMS_{lin}	RMS_{log}	σ_1	σ_2	σ_3
Initial depth [7]	8.730	9.864	0.093	0.043	0.356	0.134	0.908	0.981	**0.995**
Refined with DispField [17]	2.410	8.230	0.092	0.042	0.352	0.132	0.910	0.981	**0.995**
Refined (ours)	**1.985**	**6.990**	0.093	0.042	0.351	0.133	0.909	0.981	**0.995**
Alternative network inputs (in addition to the rough depth map)									
RGB image	8.816	9.887	0.092	0.042	0.352	0.132	0.910	0.982	**0.995**
Normal map	9.437	9.937	**0.087**	**0.038**	**0.333**	**0.125**	0.917	0.982	0.996
Binary edges	5.619	9.397	0.096	0.044	0.362	0.138	**0.902**	**0.980**	**0.995**
Different loss functions $\mathcal{L}$ exploiting the ground-truth depth									
$\mathcal{L}_{\mathsf{gtdepth}} + \mathcal{L}_{\mathsf{regul}}$	8.756	9.866	0.093	0.043	0.356	0.134	0.908	0.981	**0.995**
$\mathcal{L}_{\mathsf{occonsist}} + \mathcal{L}_{\mathsf{gtdepth}}$	2.778	8.006	0.092	0.042	0.356	0.133	0.909	0.981	**0.995**
$\mathcal{L}_{\mathsf{occonsist}} + \mathcal{L}_{\mathsf{regul}} + \mathcal{L}_{\mathsf{gtdepth}}$	3.090	7.291	0.093	0.042	0.351	0.132	0.910	0.982	**0.995**
Different depth combinations used in $\mathcal{L}_{\mathsf{occonsist}}$									
dd (order-0 depth only)	2.375	7.406	0.094	0.043	0.356	0.135	0.907	0.981	**0.995**
DD (order-1 depth only)	2.401	7.373	0.093	0.042	0.352	0.133	0.909	0.981	**0.995**

Table 9. Ablation study for refined depth map estimation: (a) using alternative network inputs, (b) using different loss functions exploiting the ground-truth depth, (c) using a different combination of order-0 and order-0 depth difference in $\mathcal{L}_{\mathsf{occonsist}}$. See details in text. Best results in **bold**.

hardly improves the quality of the output depth map, which is not surprising as most the cues that can be directly exploited from the RGB image have already been exploited by [7]. Using the normal map leads to slightly lower depth errors but much worse depth boundaries. Last, using binary occlusion edges leads to a slightly higher depth accuracy but poor depth boundaries too. In the end, our estimated occlusion relationships as guidance achieves the lowest boundary errors without a noticeable degradation or improvement of the depth error and accuracy, which we believe is the best compromise.

Different loss functions exploiting the ground-truth depth. Then we study variations in the loss function when adding ground-truth depth information at training time. We introduce the loss function $\mathcal{L}_{\mathsf{gtdepth}}$, which is the counterpart of $\mathcal{L}_{\mathsf{regul}}$ using the ground-truth depth d^{gt} instead of the rough input depth $\tilde{d}$: it penalizes the difference between the refined depth d and the ground truth depth d^{gt} as defined in Equation (10):

$$\mathcal{L}_{\mathsf{gtdepth}} = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \Big(\mathcal{B}(\log d_p^{gt}, \log d_p) + \|\nabla \log d_p^{gt} - \nabla \log d_p\|^2 \Big) \tag{10}$$

We study different combinations of partial losses, i.e., $\mathcal{L}_{\mathsf{gtdepth}} + \mathcal{L}_{\mathsf{regul}}$ (which ignores occlusion information), $\mathcal{L}_{\mathsf{occonsist}} + \mathcal{L}_{\mathsf{gtdepth}}$ (which does not penalize difference between $\tilde{d}$ and d), and $\mathcal{L}_{\mathsf{occonsist}} + \mathcal{L}_{\mathsf{regul}} + \mathcal{L}_{\mathsf{gtdepth}}$ (which combines both the rough input depth and ground-truth depth information), comparing to the loss $\mathcal{L} = \mathcal{L}_{\mathsf{occonsist}} + \mathcal{L}_{\mathsf{regul}}$ as defined in the paper (which uses only the rough input depth), i.e., line "Refined (ours)" in the table.

As shown in the middle part of Tab. 9, $\mathcal{L}_{\mathsf{gtdepth}} + \mathcal{L}_{\mathsf{regul}}$ does not improve or degrade the input depth map noticeably; information about edges [17] or occlusions is missing to yield any significant improvement. Replacing the rough input depth map by the ground truth as $\mathcal{L}_{\mathsf{occonsist}} + \mathcal{L}_{\mathsf{gtdepth}}$ significantly improves ϵ_{acc}, slightly improves ϵ_{comp} and does not affect much the general depth error and accuracy metrics. But it is not as good as with our method. Finally, using both the rough input depth map and the ground-truth depth map as $\mathcal{L}_{\mathsf{occonsist}} + \mathcal{L}_{\mathsf{regul}} + \mathcal{L}_{\mathsf{gtdepth}}$, i.e., adding ground-truth information to our setting, is not as good either as not using it.

Different combinations or order-0 and order-1 depths. Last, we also consider variations in the depths used in $\mathcal{L}_{\mathsf{occonsist}}$, cf. Eq. (8) of the paper, using either the refined order-0 depth difference d_{pq} or the tangent-adjusted order-1 depth difference D_{pq}. More precisely, we consider the cases where the signed distances in Eq. (8) are both d_{pq} (named "dd") or both D_{pq} (named "DD"), instead of d_{pq} then D_{pq} as defined in Eq. (8).

As can be seen in the bottom part of Tab. 9, the performance of both variants is not as good as the loss function we define in the paper.

C Samples of occlusion relationship in generated datasets

As described in Section 5 and Table 1 of the paper, we generated occlusion relationship annotations for three datasets, i.e., InteriorNet, iBims-1 and NYUv2. We illustrate here a few annotation samples: from InteriorNet-OR (cf. Fig. 12), iBims-1-OR (cf. Fig. 13) and NYUv2-OR (cf. Fig. 14).

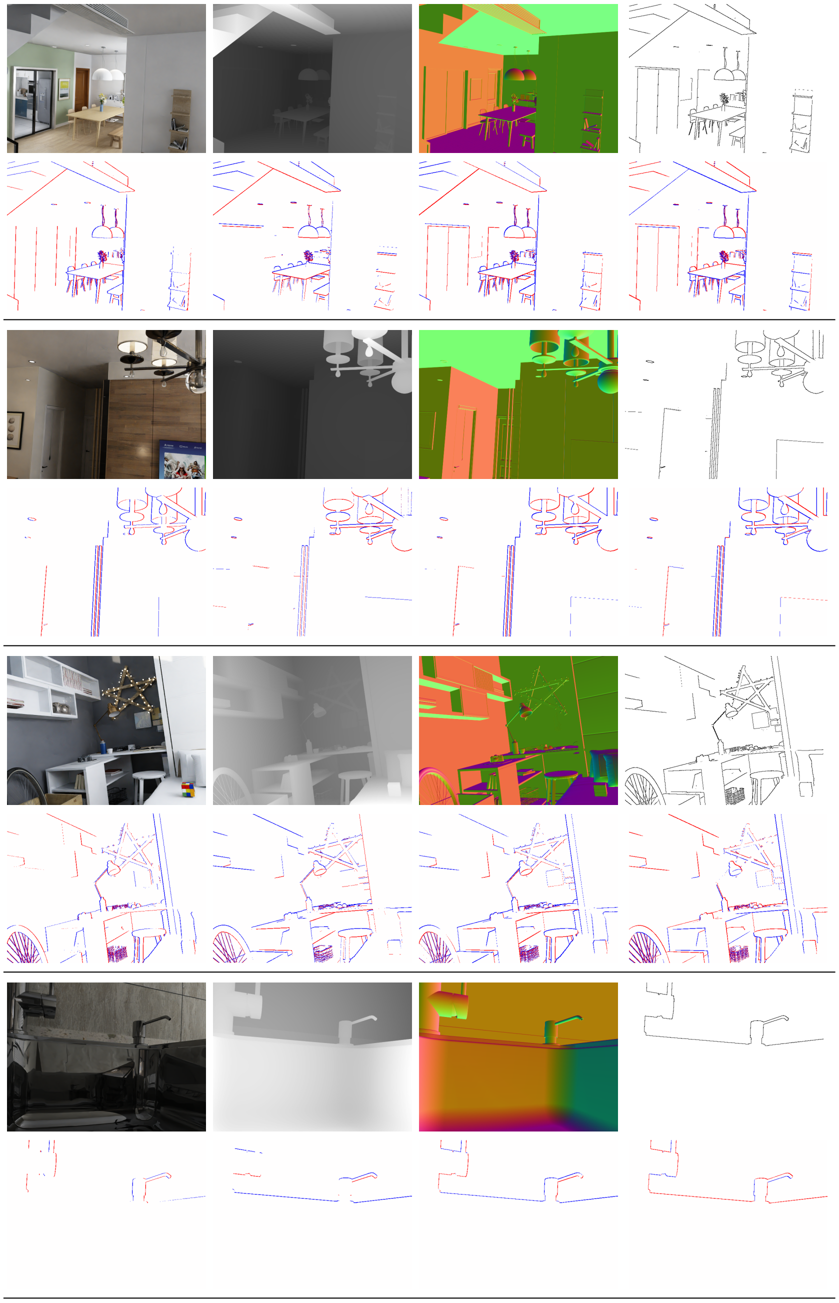

Fig. 12. Samples from our InteriorNet-OR dataset. For each sample, first row, left to right: RGB image, depth map, normal map and generated occlusion boundaries; second row, left to right: generated occlusion relationships along inclinations horizontal ($i = \mathsf{h}$), vertical ($i = \mathsf{v}$), diagonal ($i = \mathsf{d}$) and antidiagonal ($i = \mathsf{a}$). Colors blue, white and red respectively represent pixel-pair occlusion status $r = -1$, 0 or 1.

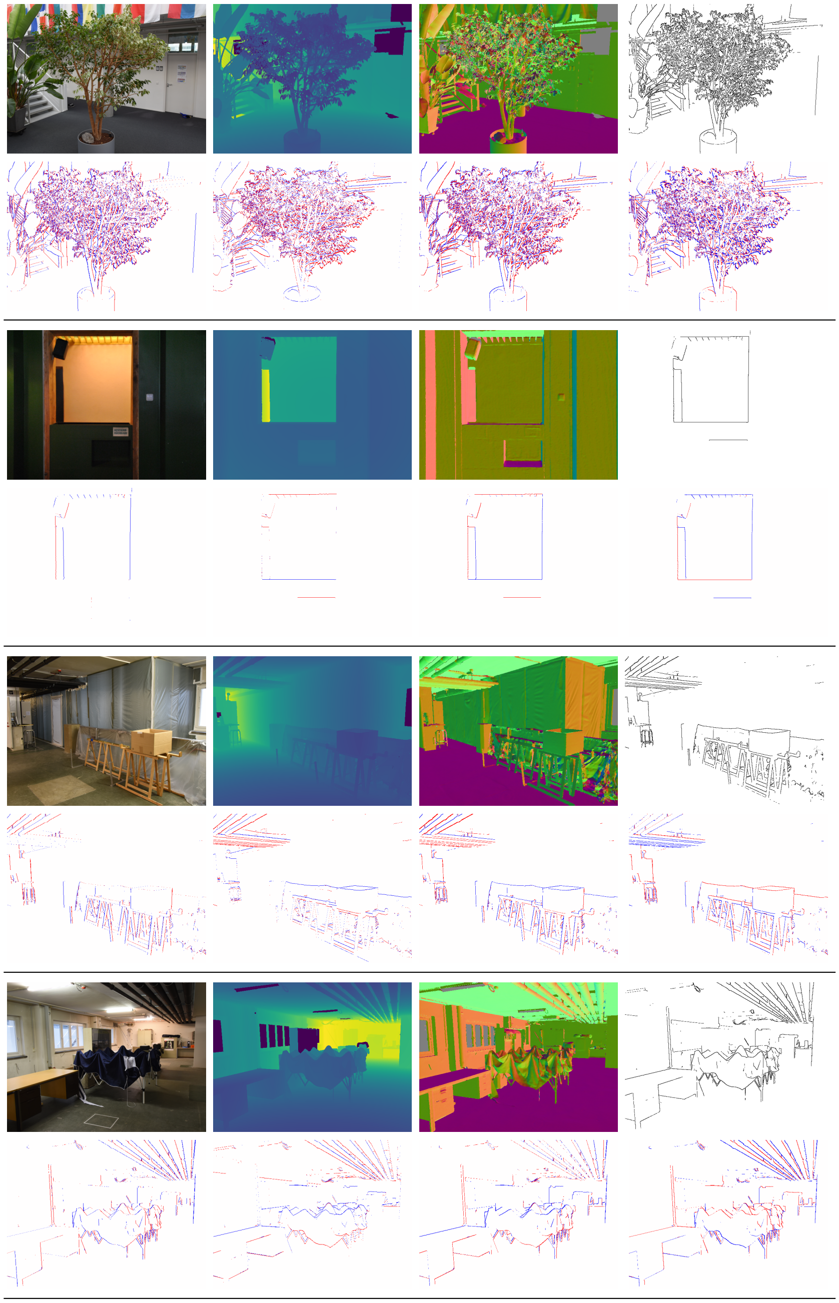

Fig. 13. Samples from our iBims-1-OR dataset. For each sample, first row, left to right: RGB image, depth map, normal map and generated occlusion boundaries; second row, left to right: generated occlusion relationships along inclinations horizontal ($i = \mathsf{h}$), vertical ($i = \mathsf{v}$), diagonal ($i = \mathsf{d}$) and antidiagonal ($i = \mathsf{a}$). Colors blue, white and red respectively represent pixel-pair occlusion status $r = -1$, 0 or 1.

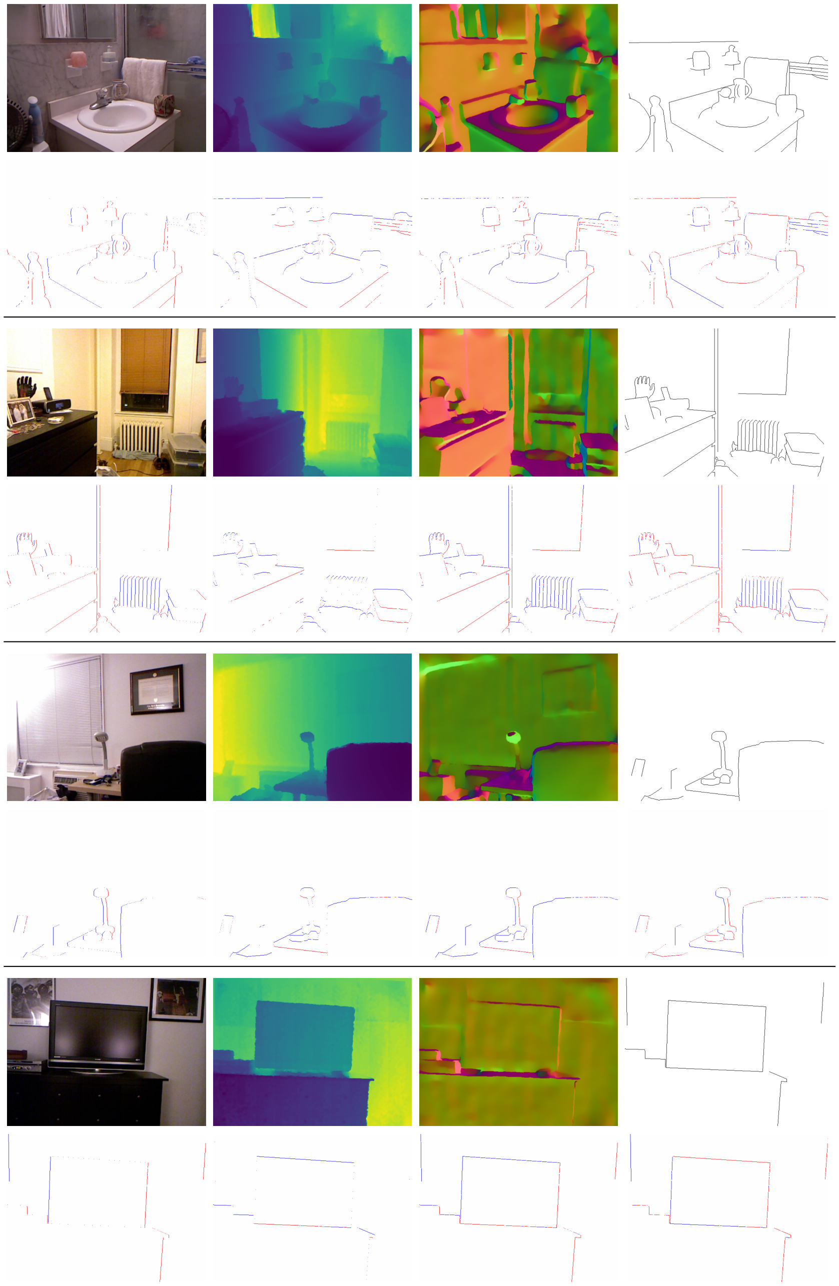

Fig. 14. Samples from our NYUv2-OR dataset. For each sample, first row, left to right: RGB image, depth map and occlusion boundaries labeled by [17]; second row, left to right: generated occlusion relationships along inclinations horizontal ($i = \mathsf{h}$), vertical ($i = \mathsf{v}$), diagonal ($i = \mathsf{d}$) and antidiagonal ($i = \mathsf{a}$). Colors blue, white and red respectively represent pixel-pair occlusion status $r = -1$, 0 or 1.

References

1. Barron, J.T., Poole, B.: The fast bilateral solver. In: European Conference on Computer Vision (ECCV). pp. 617–632 (2016)
2. Boulch, A., Marlet, R.: Fast and robust normal estimation for point clouds with sharp features. Computer Graphics Forum (CGF) **31**(5), 1765–1774 (2012)
3. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems (NeurIPS), pp. 2366–2374. Curran Associates, Inc. (2014)
4. Fu, H., Gong, M., Wang, C., Batmanghelich, K., Tao, D.: Deep ordinal regression network for monocular depth estimation. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2002–2011 (2018)
5. He, K., Sun, J., Tang, X.: Guided image filtering. In: European Conference on Computer Vision (ECCV). pp. 1–14 (2010)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)
7. Jiao, J., Cao, Y., Song, Y., Lau, R.W.H.: Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss. In: European Conference on Computer Vision (ECCV) (2018)
8. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
9. Koch, T., Liebel, L., Fraundorfer, F., Körner, M.: Evaluation of cnn-based single-image depth estimation methods. In: Leal-Taix, L., Roth, S. (eds.) European Conference on Computer Vision Workshops (ECCV Workshops). pp. 331–348. Springer International Publishing (2019)
10. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper depth prediction with fully convolutional residual networks. In: International Conference on 3D Vision (3DV). pp. 239–248. IEEE (2016)
11. Li, J.Y., Klein, R., Yao, A.: A two-streamed network for estimating fine-scaled depth maps from single rgb images. International Conference on Computer Vision (ICCV) pp. 3392–3400 (2016)
12. Li, W., Saeedi, S., McCormac, J., Clark, R., Tzoumanikas, D., Ye, Q., Huang, Y., Tang, R., Leutenegger, S.: Interiornet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. In: British Machine Vision Conference (BMVC) (2018)
13. Liu, C., Yang, J., Ceylan, D., Yumer, E., Furukawa, Y.: Planenet: Piece-wise planar reconstruction from a single rgb image. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2579–2588 (2018)
14. Liu, F., Shen, C., Lin, G.: Deep convolutional neural fields for depth estimation from a single image. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
15. Lu, R., Xue, F., Zhou, M., Ming, A., Zhou, Y.: Occlusion-shared and feature-separated network for occlusion relationship reasoning. In: International Conference on Computer Vision (ICCV) (2019)
16. Nathan Silberman, Derek Hoiem, P.K., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: European Conference on Computer Vision (ECCV) (2012)
17. Ramamonjisoa, M., Du, Y., Lepetit, V.: Predicting sharp and accurate occlusion boundaries in monocular depth estimation using displacement fields. arXiv preprint arXiv:2002.12730 (2020)

18. Ramamonjisoa, M., Lepetit, V.: Sharpnet: Fast and accurate recovery of occluding contours in monocular depth estimation. In: International Conference on Computer Vision Workshops (ICCV Workshops) (2019)
19. Su, H., Jampani, V., Sun, D., Gallo, O., Learned-Miller, E., Kautz, J.: Pixel-adaptive convolutional neural networks. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11166–11175 (2019)
20. Teo, C., Fermuller, C., Aloimonos, Y.: Fast 2D border ownership assignment. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5117–5125 (2015)
21. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: International Conference on Computer Vision (ICCV). pp. 839–846 (1998)
22. Wang, G., Liang, X., Li, F.W.B.: DOOBNet: Deep object occlusion boundary detection from an image. In: Asian Conference on Computer Vision (ACCV) (2018)
23. Wang, P., Yuille, A.: DOC: Deep occlusion estimation from a single image. In: European Conference on Computer Vision (ECCV) (2016)
24. Wu, H., Zheng, S., Zhang, J., Huang, K.: Fast end-to-end trainable guided filter. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1838–1847 (2018)
25. Yin, W., Liu, Y., Shen, C., Yan, Y.: Enforcing geometric constraints of virtual normal for depth prediction. In: International Conference on Computer Vision (ICCV) (2019)
26. Zheng, C., Cham, T.J., Cai, J.: T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks. In: European Conference on Computer Vision (ECCV). pp. 767–783 (2018)