IronDepth: Iterative Refinement of Single-View Depth using Surface Normal and its Uncertainty

Gwangbin Bae, Ignas Budvytis, Roberto Cipolla

University of Cambridge

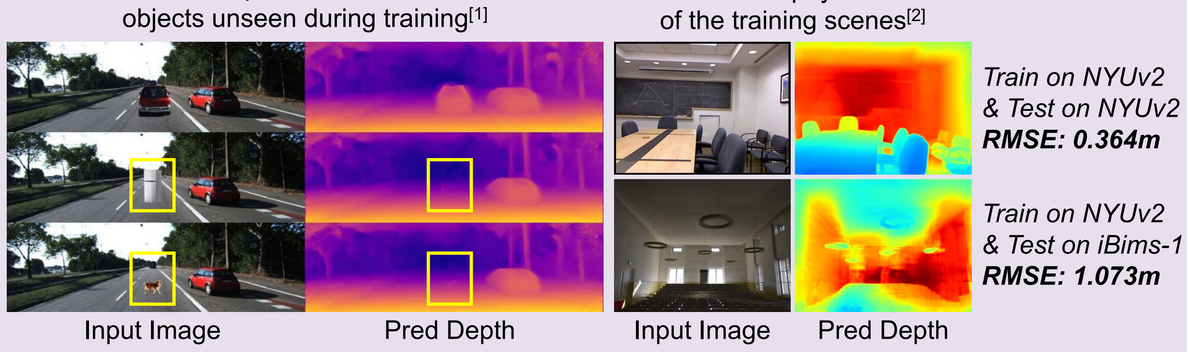
Motivation

Limitations of existing depth estimation methods:

1) Poor generalization ability

Poor performance for objects unseen during training[1]

Biased to the physical scale of the training scenes[2]



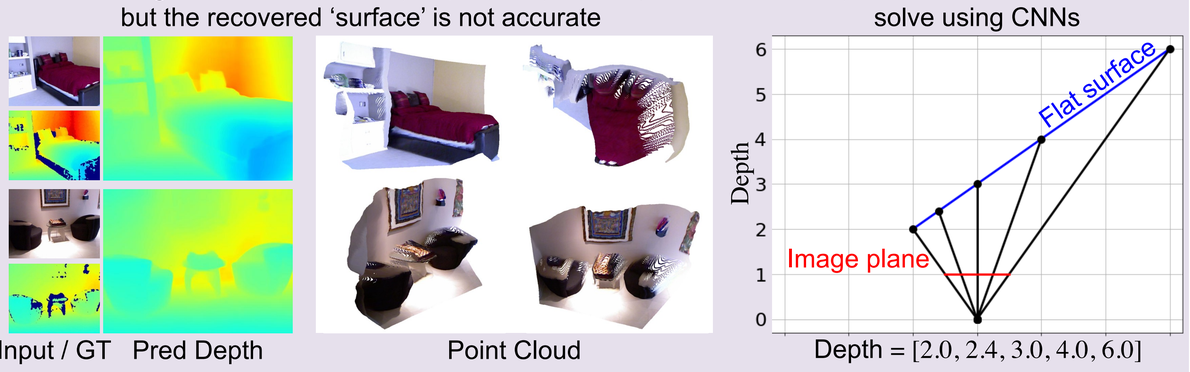
Train on NYUv2 & Test on NYUv2
RMSE: 0.364m

Train on NYUv2 & Test on iBims-1
RMSE: 1.073m

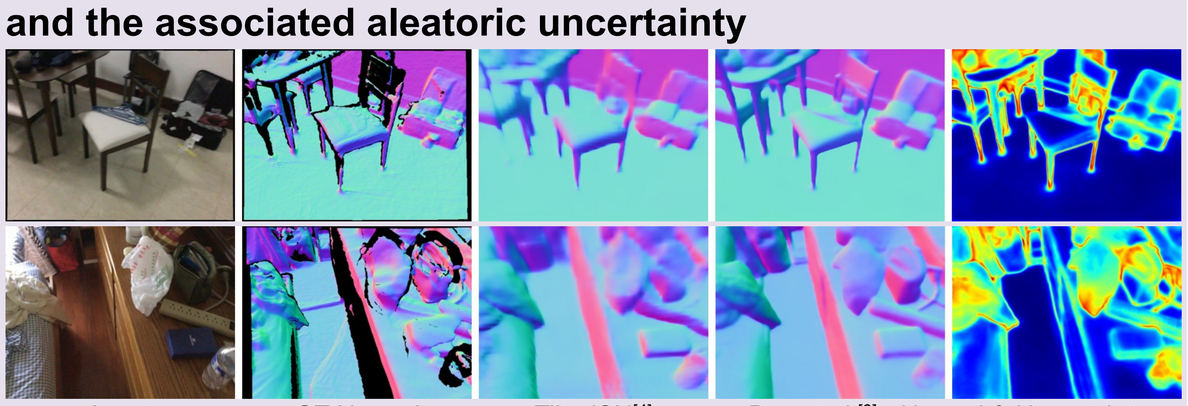
2) Poor surface normal accuracy

Existing methods show low 'per-pixel' depth error, but the recovered 'surface' is not accurate

Depth estimation is difficult to solve using CNNs



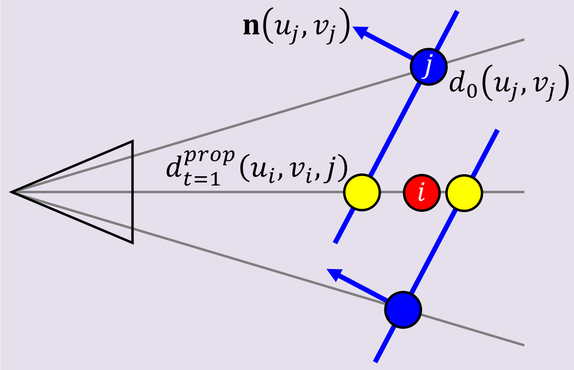
- In our previous work[3], **we estimated per-pixel surface normal and the associated aleatoric uncertainty**



Method

Normal-guided depth refinement

- Depth of a pixel can be propagated to a target pixel, using the predicted surface normal as guidance
- We can formulate depth refinement as classification of choosing the neighboring pixel to propagate from



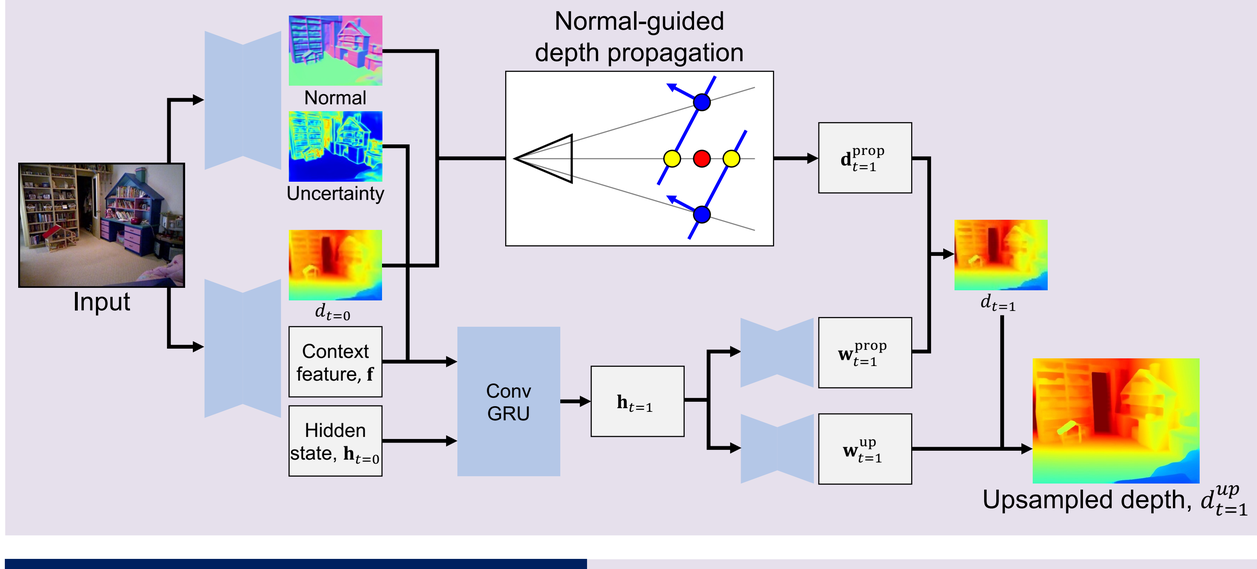
$$\mathbf{r}(u_i, v_i) = \begin{pmatrix} \frac{u_i - u_0}{f_u} & \frac{v_i - v_0}{f_v} & 1 \end{pmatrix}^\top$$

$$d_{t=1}^{prop}(u_i, v_i, j) = \frac{\mathbf{n}(u_j, v_j) \cdot \mathbf{r}(u_j, v_j)}{\mathbf{n}(u_j, v_j) \cdot \mathbf{r}(u_i, v_i)} \cdot d_0(u_j, v_j)$$

$$d_{t=1}(u_i, v_i) = \sum_{j \in \mathcal{N}_i} w_{t=1}^{prop}(u_i, v_i, j) \cdot d_{t=1}^{prop}(u_i, v_i, j)$$

IronDepth pipeline

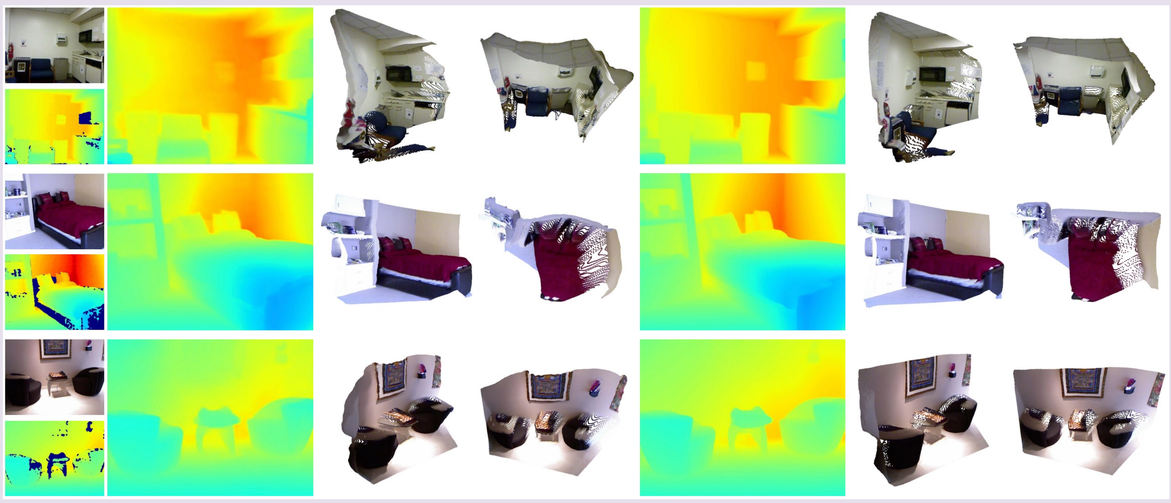
- Iteratively refine the coarse resolution depth-map using normal-guided depth propagation
- Upsample the refined depth-map by applying the normal-guided propagation to sub-pixel points



Results

1) Qualitative comparison

- Our method is better at capturing the orientation of the surfaces



2) Quantitative comparison

- Results on NYUv2

Method	Depth error			Depth accuracy			Normal error			Normal accuracy		
	abs rel	rmse	$\log_{10}$	δ_1	δ_2	δ_3	mean	median	rmse	11.25°	22.5°	30°
DORN [5]	0.106	0.397	0.046	0.877	0.970	0.990	44.7	39.3	53.3	9.2	26.7	38.0
VNL [6]	**0.100**	0.368	**0.043**	0.895	0.980	0.996	26.8	17.0	37.9	36.3	59.4	68.6
BTS [7]	0.110	0.392	0.047	0.886	0.978	0.994	32.4	24.7	42.1	22.7	46.1	58.3
AdaBins [2]	0.103	0.364	0.044	0.902	0.983	**0.997**	28.8	20.7	38.6	28.3	53.2	64.7
Ours	0.101	**0.352**	**0.043**	**0.910**	**0.985**	**0.997**	**20.8**	**11.3**	**31.9**	**49.7**	**70.5**	**77.9**

- Results on iBims-1 (network trained on NYUv2)

Method	Depth error			Depth accuracy			Normal error			Normal accuracy			Planarity	
	rel	rmse	$\log_{10}$	δ_1	δ_2	δ_3	mean	median	rmse	11.25°	22.5°	30°	ε^{plan}	ε^{orie}
VNL [6]	0.24	1.07	0.11	0.55	0.85	0.94	39.8	30.4	51.0	17.9	38.6	49.4	6.49	18.72
BTS [7]	0.24	1.08	0.12	0.53	0.84	0.94	44.0	37.8	53.5	13.0	29.5	40.0	7.25	20.52
DAV [8]	0.24	1.06	**0.10**	**0.59**	0.84	0.94	-	-	-	-	-	-	7.21	18.45
AdaBins [2]	0.22	1.06	0.11	0.55	0.86	**0.95**	37.1	29.6	46.9	18.0	38.7	50.6	6.25	17.51
Ours	**0.21**	**1.03**	0.11	**0.59**	**0.87**	**0.95**	**25.3**	**14.2**	**37.4**	**43.1**	**63.9**	**71.6**	**3.29**	**8.48**

Usefulness of IronDepth

1) IronDepth can improve the accuracy of existing methods

Method	Depth error			Depth accuracy			Normal error			Normal accuracy		
	abs rel	rmse	$\log_{10}$	δ_1	δ_2	δ_3	mean	median	rmse	11.25°	22.5°	30°
DORN	0.106	0.397	0.046	0.877	0.970	0.990	44.7	39.3	53.3	9.2	26.7	38.0
DORN + Ours	**0.099**	**0.359**	**0.042**	**0.898**	**0.978**	**0.993**	**21.3**	**11.8**	**32.5**	**48.5**	**69.6**	**77.1**
VNL	0.100	0.368	0.043	0.895	0.980	0.996	26.8	17.0	37.9	36.3	59.4	68.6
VNL + Ours	**0.097**	**0.353**	**0.042**	**0.902**	**0.983**	**0.996**	**20.5**	**11.0**	**31.7**	**50.6**	**71.0**	**78.2**
AdaBins	0.103	0.364	0.044	0.902	0.983	0.997	28.8	20.7	38.6	28.3	53.2	64.7
AdaBins + Ours	**0.100**	**0.351**	**0.042**	**0.911**	**0.985**	**0.997**	**20.7**	**11.3**	**31.8**	**49.9**	**70.6**	**78.0**

2) IronDepth can seamlessly be applied to depth completion

# Measurements	Depth metrics (w/o scale-match)						Depth metrics (w/ scale-match)					
	abs rel	rmse	$\log_{10}$	δ_1	δ_2	δ_3	abs rel	rmse	$\log_{10}$	δ_1	δ_2	δ_3
0	0.101	0.352	0.043	0.910	0.985	0.997	0.101	0.352	0.043	0.910	0.985	0.997
10	0.097	0.341	0.041	0.917	0.986	0.997	0.076	0.300	0.033	0.944	0.991	0.998
50	0.084	0.304	0.035	0.938	0.990	0.998	0.063	0.260	0.027	0.962	0.994	0.999
100	0.070	0.266	0.030	0.957	0.993	0.999	0.053	0.231	0.023	0.972	0.995	0.999
200	0.051	0.212	0.021	0.976	0.996	0.999	0.041	0.191	0.018	0.983	0.997	0.999

Project Page

Demo (YouTube)

Code (GitHub)

References

[1] How do neural networks see depth in single images?, Dijk and Croon, ICCV 2019
[2] Adabins: Depth estimation using adaptive bins, Bhat et al., CVPR 2021
[3] Estimating and exploiting the aleatoric uncertainty in surface normal estimation, Bae et al., ICCV 2021
[4] Surface normal estimation of tilted images via spatial rectifier, Do et al., ECCV 2020
[5] Deep ordinal regression network for monocular depth estimation, Fu et al., CVPR 2018
[6] Enforcing geometric constraints of virtual normal for depth prediction, Yin et al., ICCV 2019
[7] From big to small: Multi-scale local planar guidance for monocular depth estimation, Lee et al., arXiv 2019
[8] Guiding monocular depth estimation using depth-attention volume, Huynh et al., ECCV 2020

UNIVERSITY OF CAMBRIDGE

BMVC 2022