Unsupervised Learning of Depth and Ego-Motion from Video*

Bucharest Computer Vision Reading Group

Presented by Emanuela Haller

*Tinghui Zhou, Matthew Brown, Noah Snavely and David G. Lowe
UC Berkely, Google
CVPR 2017

Introduction

- Input:
  - Unstructured video sequences
- Output:
  - Depth map
    - Monocular observation
  - Ego-motion
    - Camera motion relative to a rigid scene
    - 6 DoF

- Training:
  - Unsupervised

- Results:
  - Monocular depth – comparably with supervised methods
  - Pose estimation – favorably comparable to established SLAM systems under comparable input settings

Motivation

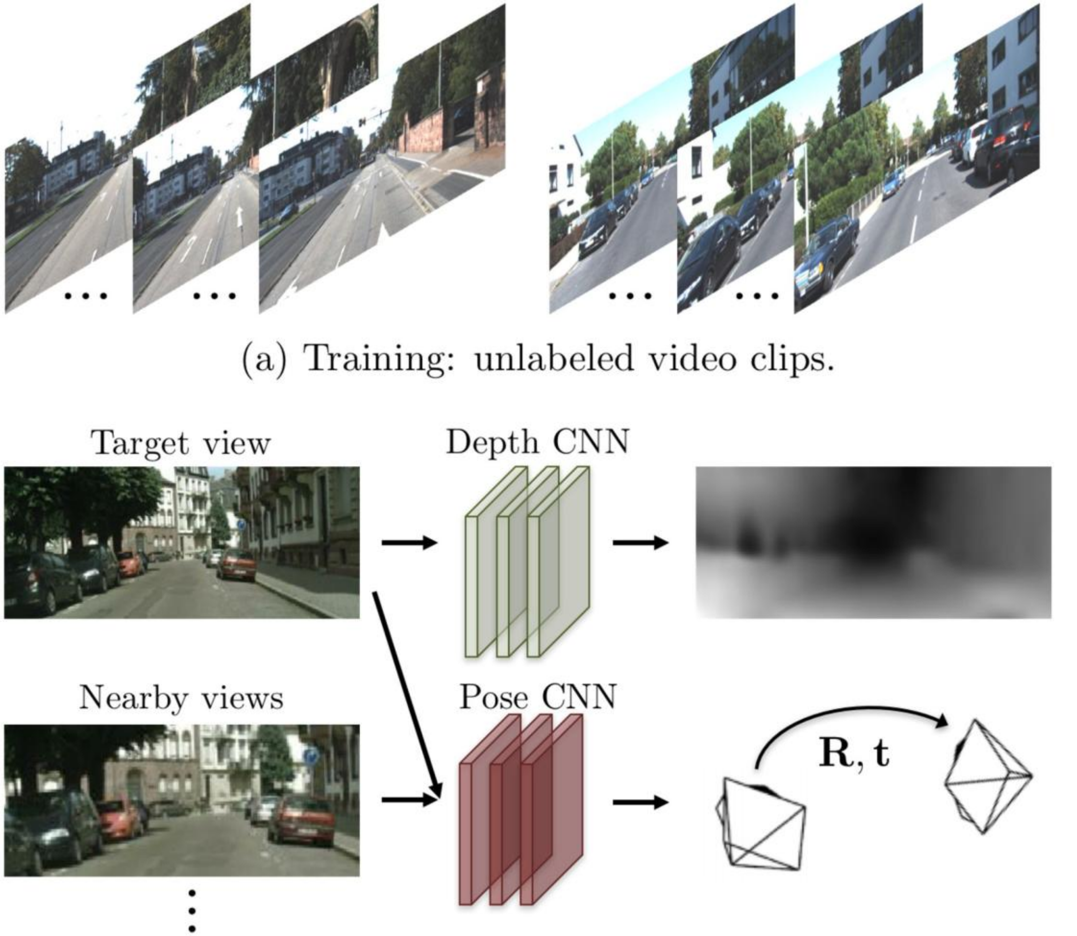
- Simulate human performances of inferring ego-motion and the 3D structure of a scene even over short timescales

- Why do humans excel at this task?
  - Development of rich, structural understanding of the world through our past visual experience
  - Learn regularities of the world

- Train a model that observes sequences of images and aims to explain its observations by predicting likely camera motion and the scene structure

- Meta-task – view synthesis

  => Learn intermediate tasks (depth and camera pose estimation)



(a) Training: unlabeled video clips.

(b) Testing: single-view depth and multi-view pose estimation.

Assumptions

- Ideal situation
  - The scene is static, without moving objects
    - Changes are dominated by camera motion
  - There is no occlusion/disocclusion between source and target views
  - The surface is Lambertian so that no photo-consistency error is meaningful
- Handle model limitations
  - Explainability prediction network

Approach

- Jointly train:
  - Single-view depth CNN
  - Camera pose estimation CNN

- Supervision signal: view synthesis

$$\left.\begin{matrix} per-pixel\ depth\ map\ of\ target \\ pose \\ visibility\ in\ nearby\ view \end{matrix}\right\} => target\ view$$

- Explainability prediction network
  - jointly and simultaneously with depth and pose networks

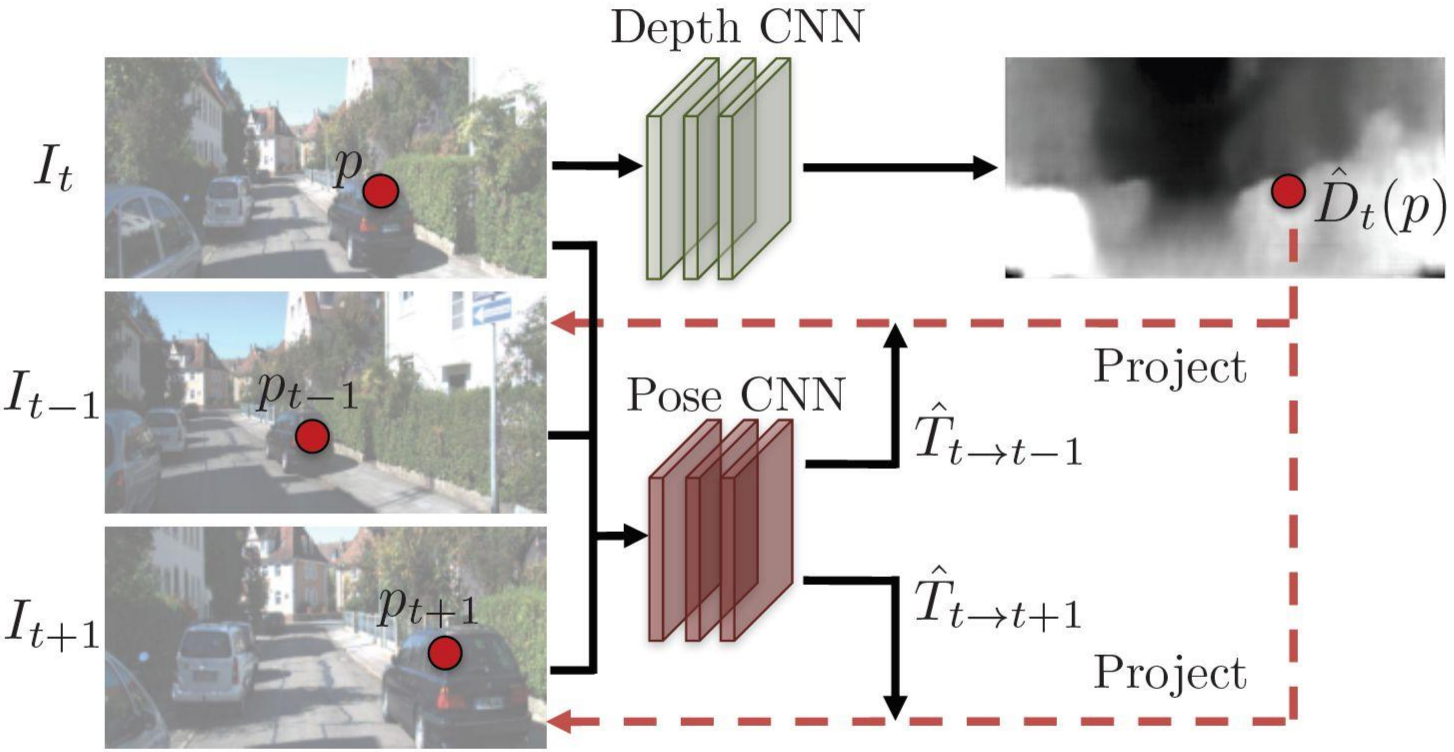
View synthesis as supervision

- Previous approaches
  - Single view depth estimation
    - "Unsupervised CNN for single view depth estimation: Geometry to the rescue" – ECCV 2016

      R. Garg, V.K. BG, G Carneiro and I. Reid
    - "Unsupervised monocular depth estimation with left-right consistency" – CVPR 2017

      C Godard, O. Mac Aodha and G.J. Brostow
  - Multi-view stereo
    - "DeepStereo: Learning to Predict New Views from the World's Imagery" – CVPR 2016

      J. Flynn, I. Neulander, J. Philbin and N. Sanvely
- Previous work requires posed image sets during training



$< I_1, I_2, \ldots I_N >$ −training sequence

I_t – target view

I_s $(1 \le s \le N, s \ne t)$ − source views

$\mathcal{L}_{vs} = \sum_s \sum_p |I_t(p) - \hat{I}_s(p)|$

$\hat{I}_s$ − I_s warped to the target coordinate frame

$\widehat{D}_t$ − predicted depth

$\hat{T}_{t\to s}$ − transformation matrix

$\hat{I}_s = f(I_s;\ \widehat{D}_t, \hat{T}_{t\to s})$

$$\hat{I}_s = f(I_s;\ \hat{D}_t, \hat{T}_{t\to s})$$

Obtain $I_s(p_s)$ and populate $\hat{I}_s(p_t)$

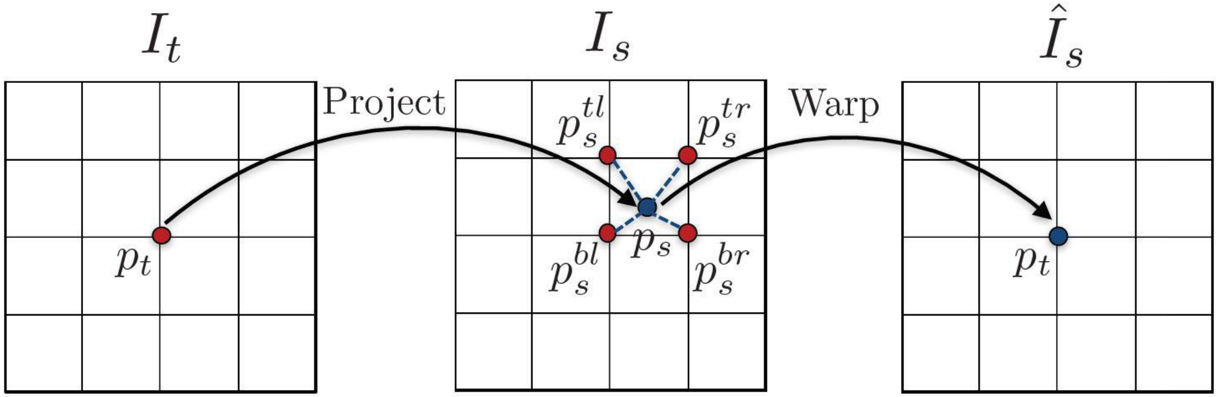
p_t – coordinates of a pixel in the target view
p_s – coordinates of p_t projected onto the source view

$$p_s \sim K\hat{T}_{t\to s}\hat{D}_t(p_t)K^{-1}p_t$$

K – camera intrinsics matrix

$$\hat{I}_s(p_t) = I_s(p_s) = \sum_{i\in\{t,b\},j\in\{l,r\}} w^{ij} I_s(p_s^{ij})$$

$$\sum_{i,j} w^{ij} = 1$$



Explainability prediction network

$\hat{E}_s -$ per pixel soft mask

- Network’s belief in where direct view synthesis will be successfully modeled for each target pixel

$$\mathcal{L}_{vs} = \sum_s \sum_p \hat{E}_s(p) |I_t(p) - \hat{I}_s(p)|$$

- Avoid trivial solution

$$\mathcal{L}_{reg}(\hat{E}_s)$$

Overcoming gradient locality

$$\text{gradients} - f(I_t(p_t) - neigh(I_s(p_s)))$$

inhibit training if ground truth p_s in low texture region or far from current location

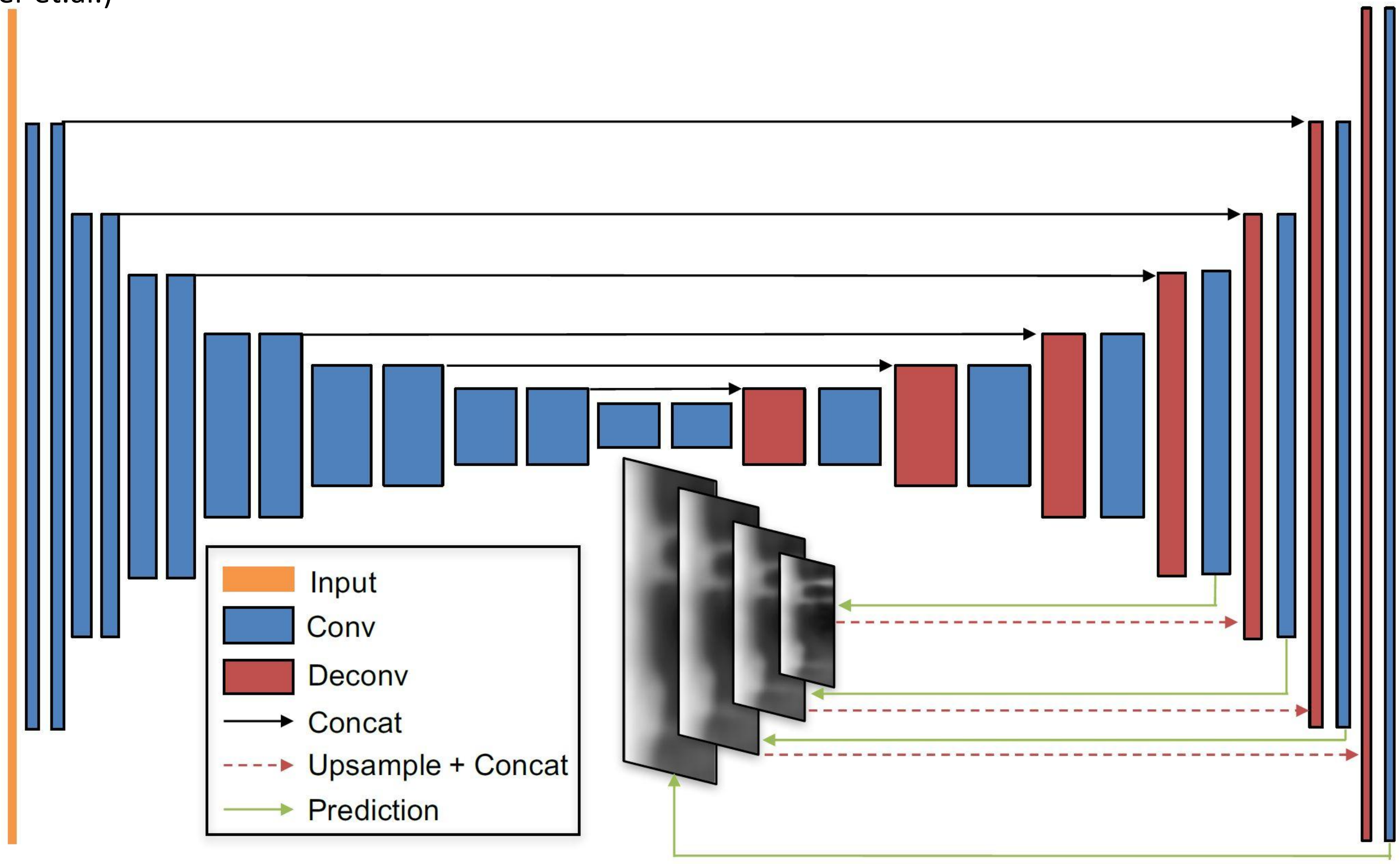
- Explicit multi-scale and smoothness loss – gradients derived from larger spatial regions

$$\mathcal{L} = \sum_l \mathcal{L}_{vs}^l + \lambda_s \mathcal{L}_{smooth}^l + \lambda_e \sum_s \mathcal{L}_{reg}(\hat{E}_s^l)$$

l – over image scales

s – over source views

“A large dataset to train convolutional networks for disparity, optical flow and scene flow estimation” – CVPR’16 (N. Mayer et.al.)

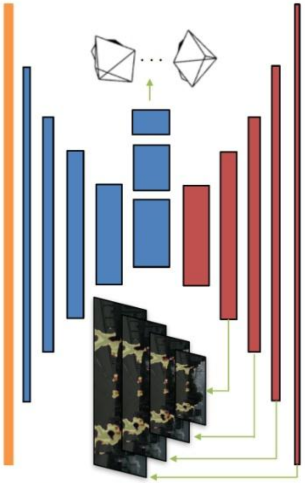


Single-view depth network

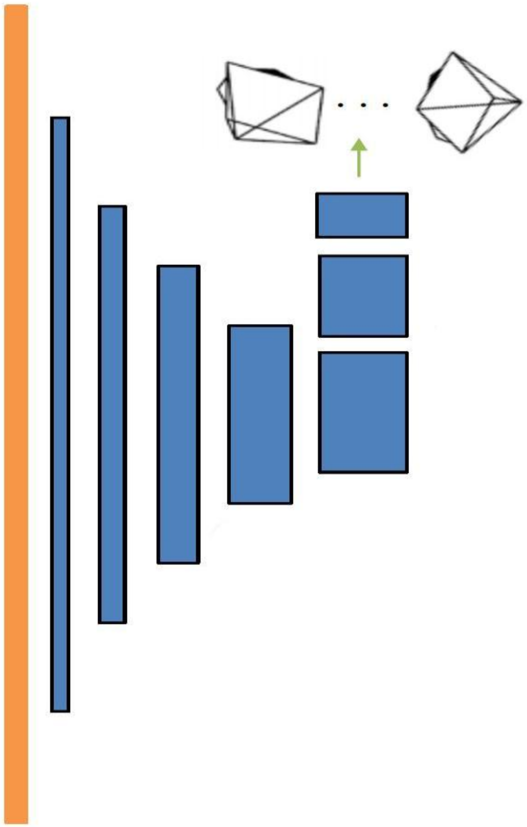
Multi-scale side predictions

ReLU activations

except for prediction layer $\frac{1}{(\alpha * sigmoid(x) + \beta)}$ $\alpha = 10, \beta = 0.1$



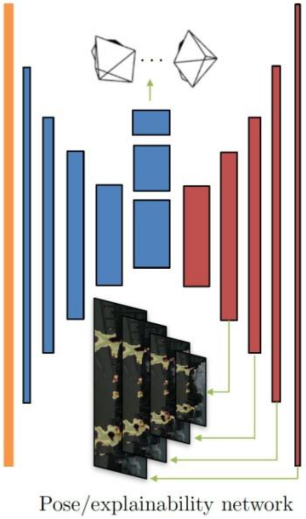
Pose/explainability network

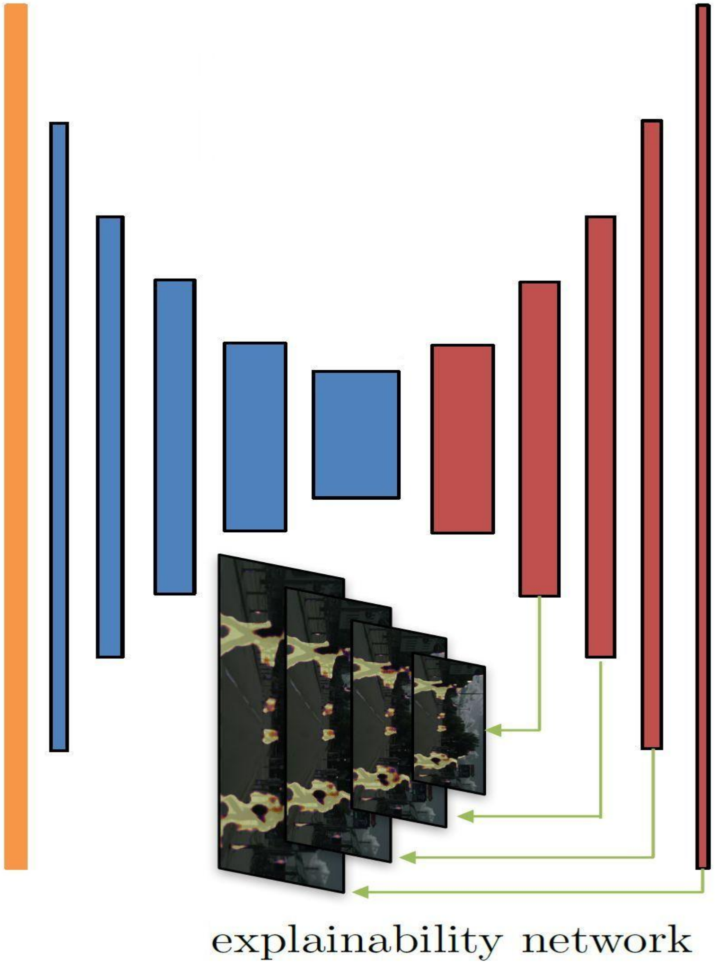


Pose network

ReLU activations except for prediction layer

6*(N-1) outputs





Multi-scale side predictions

ReLU activations except for prediction layer

2*(N-1) outputs (softmax normalization => $\hat{E}_s$)

Training details

- TensorFlow

- Depth
  - Cityscapes
  - Cityscapes + KITTI
  - Make3D
- Pose
  - KITTI

- Single-view depth estimation
  - 3 frames
- Pose estimation
  - 5 frames

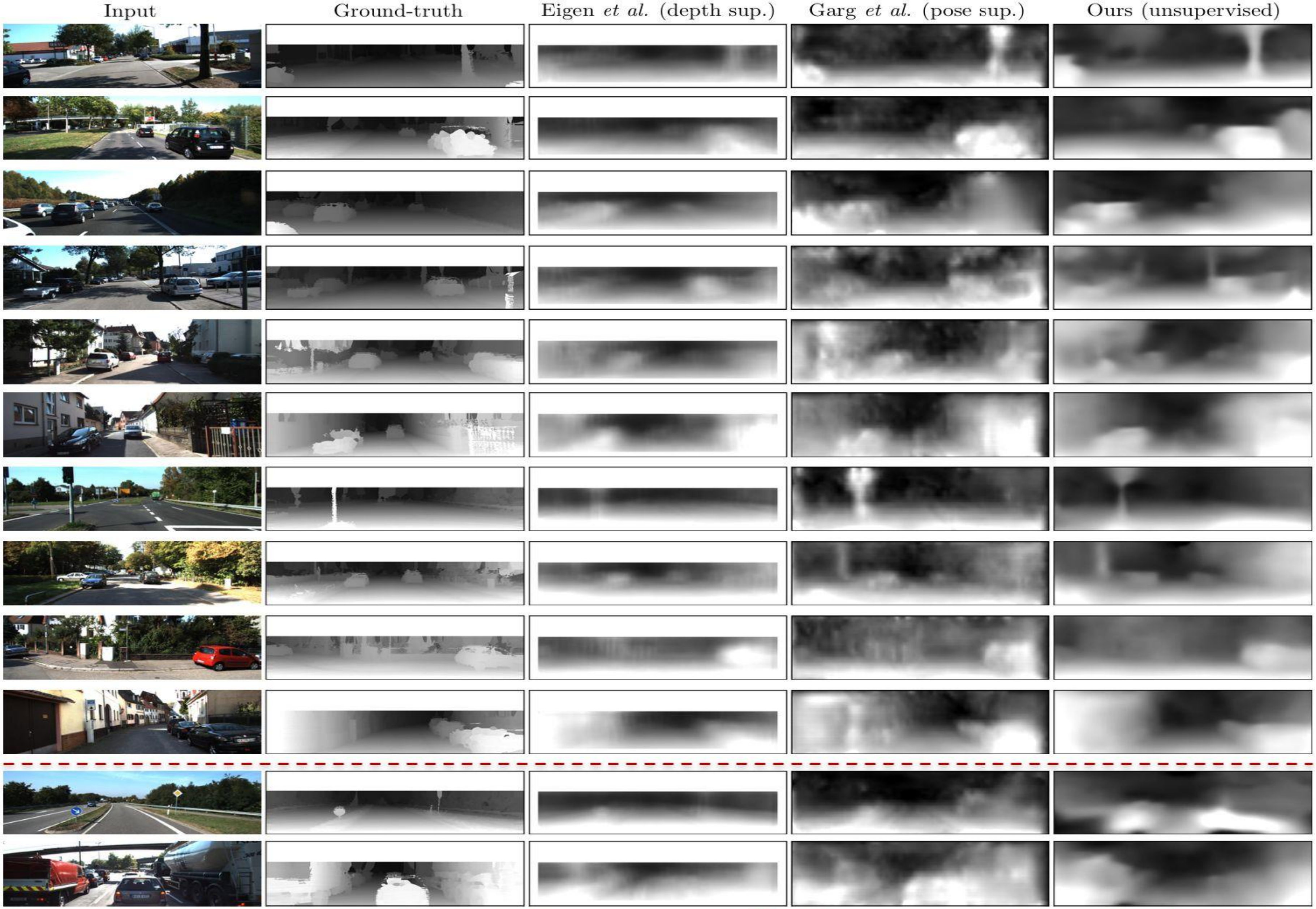


Figure 6. Comparison of single-view depth estimation between Eigen *et al.* [7] (with ground-truth depth supervision), Garg *et al.* [14] (with ground-truth pose supervision), and ours (unsupervised). The ground-truth depth map is interpolated from sparse measurements for visualization purpose. The last two rows show typical failure cases of our model, which sometimes struggles in vast open scenes and objects close to the front of the camera.

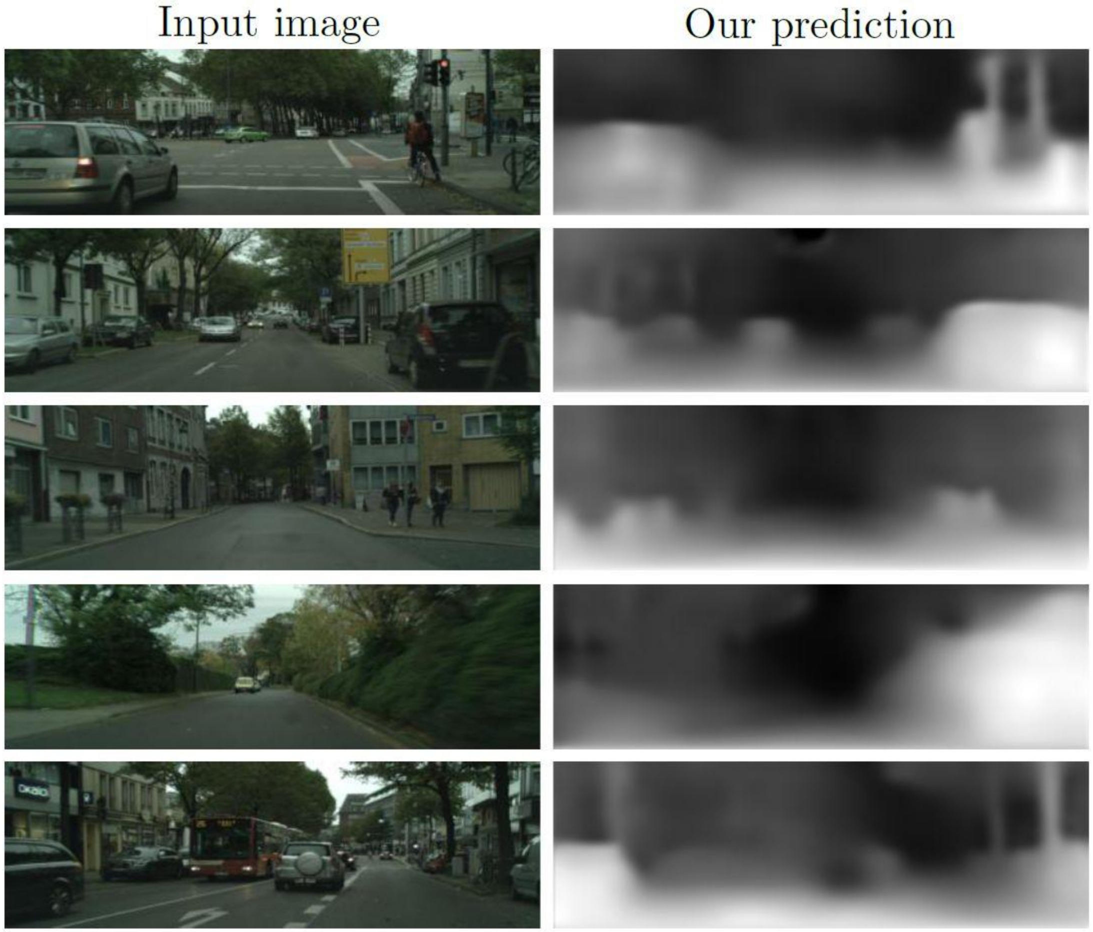


Figure 5. Our sample predictions on the Cityscapes dataset using the model trained on Cityscapes only.

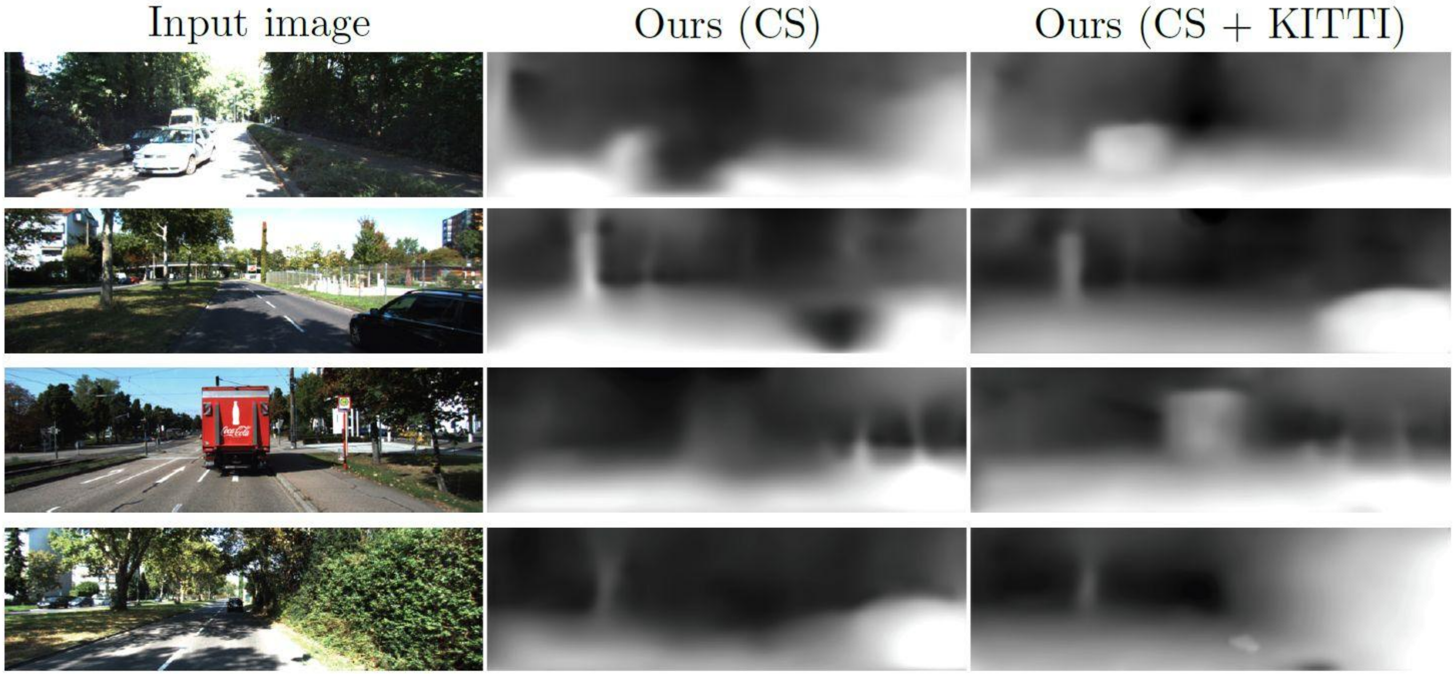


Figure 7. Comparison of single-view depth predictions on the KITTI dataset by our initial Cityscapes model and the final model (pre-trained on Cityscapes and then fine-tuned on KITTI). The Cityscapes model sometimes makes structural mistakes (e.g. holes on car body) likely due to the domain gap between the two datasets.

Method	Dataset	Supervision		Error metric				Accuracy metric		
		Depth	Pose	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Train set mean	K	✓		0.403	5.530	8.709	0.403	0.593	0.776	0.878
Eigen *et al.* [7] Coarse	K	✓		0.214	1.605	6.563	0.292	0.673	0.884	0.957
Eigen *et al.* [7] Fine	K	✓		0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu *et al.* [32]	K	✓		0.202	1.614	6.523	0.275	0.678	0.895	0.965
Godard *et al.* [16]	K		✓	0.148	1.344	5.927	0.247	0.803	0.922	0.964
Godard *et al.* [16]	CS + K		✓	0.124	1.076	5.311	0.219	0.847	0.942	0.973
Ours (w/o explainability)	K			0.221	2.226	7.527	0.294	0.676	0.885	0.954
Ours	K			0.208	1.768	6.856	0.283	0.678	0.885	0.957
Ours	CS			0.267	2.686	7.580	0.334	0.577	0.840	0.937
Ours	CS + K			0.198	1.836	6.565	0.275	0.718	0.901	0.960
Garg *et al.* [14] cap 50m	K		✓	0.169	1.080	5.104	0.273	0.740	0.904	0.962
Ours (w/o explainability) cap 50m	K			0.208	1.551	5.452	0.273	0.695	0.900	0.964
Ours cap 50m	K			0.201	1.391	5.181	0.264	0.696	0.900	0.966
Ours cap 50m	CS			0.260	2.232	6.148	0.321	0.590	0.852	0.945
Ours cap 50m	CS + K			0.190	1.436	4.975	0.258	0.735	0.915	0.968

Table 1. Single-view depth results on the KITTI dataset [15] using the split of Eigen *et al.* [7] (Baseline numbers taken from [16]). For training, K = KITTI, and CS = Cityscapes [5]. All methods we compare with use some form of supervision (either ground-truth depth or calibrated camera pose) during training. Note: results from Garg et al. [14] are capped at 50m depth, so we break these out separately in the lower part of the table.

Threshold: $\%$ of y_i s.t. $\max(\frac{y_i}{y_i^*}, \frac{y_i^*}{y_i}) = \delta < thr$

Abs Relative difference: $\frac{1}{|T|}\sum_{y \in T} |y - y^*|/y^*$

Squared Relative difference: $\frac{1}{|T|}\sum_{y \in T} ||y - y^*||^2/y^*$

RMSE (linear): $\sqrt{\frac{1}{|T|}\sum_{y \in T} ||y_i - y_i^*||^2}$

RMSE (log): $\sqrt{\frac{1}{|T|}\sum_{y \in T} ||\log y_i - \log y_i^*||^2}$

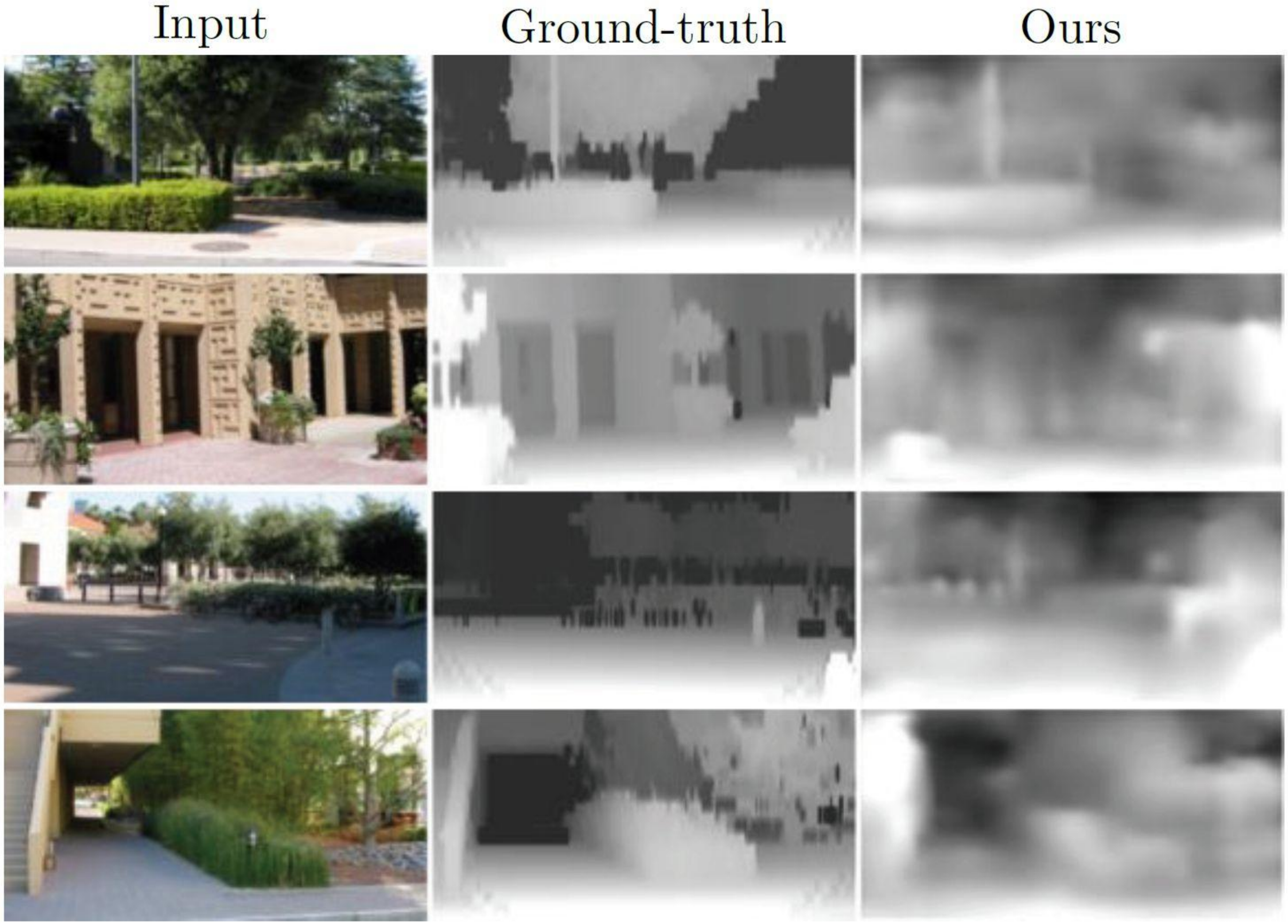
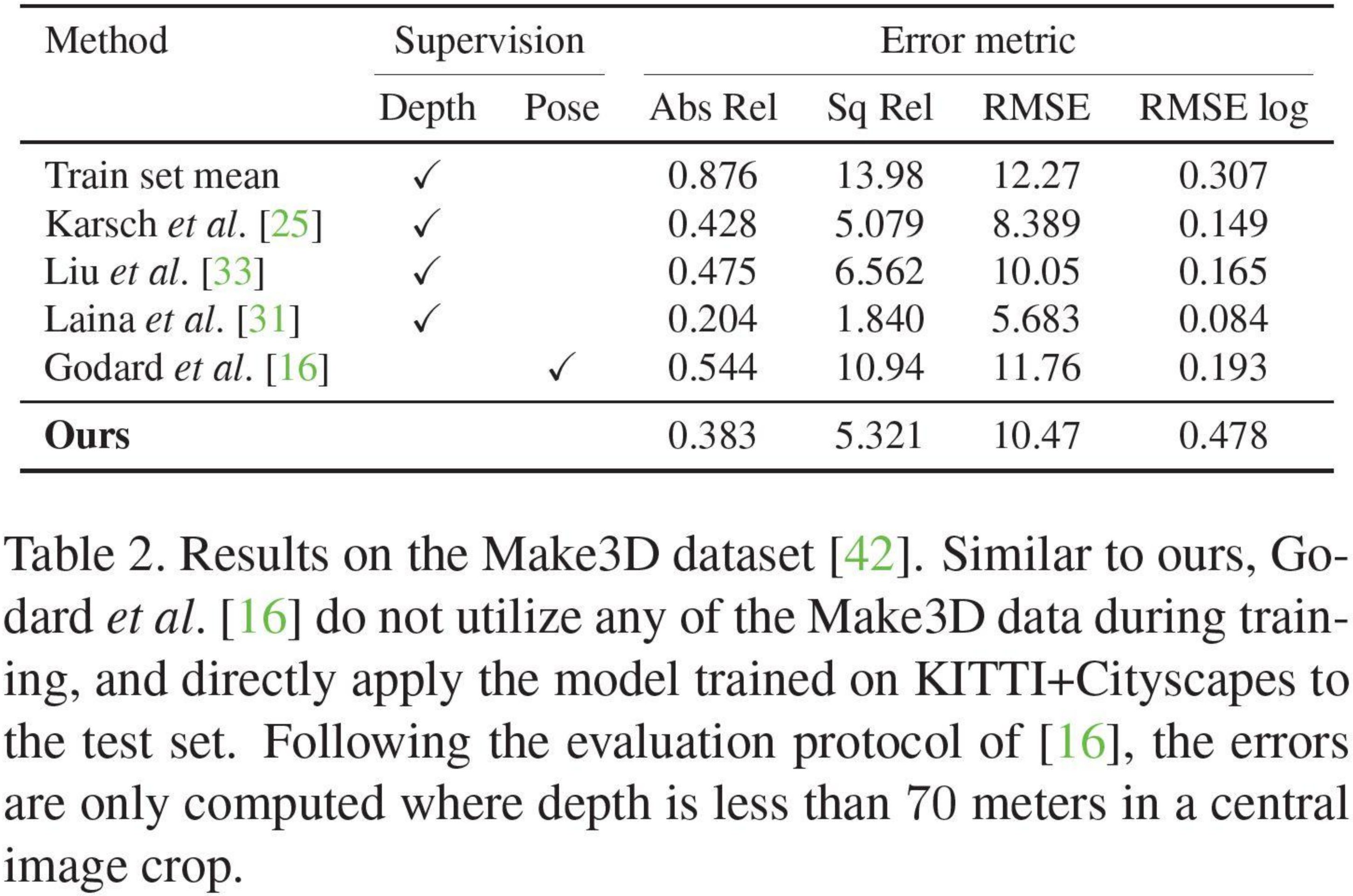


Figure 8. Our sample predictions on the Make3D dataset. Note that our model is trained on KITTI + Cityscapes only, and directly tested on Make3D.



Method	Supervision		Error metric			
	Depth	Pose	Abs Rel	Sq Rel	RMSE	RMSE log
Train set mean	✓		0.876	13.98	12.27	0.307
Karsch *et al*. [25]	✓		0.428	5.079	8.389	0.149
Liu *et al*. [33]	✓		0.475	6.562	10.05	0.165
Laina *et al*. [31]	✓		0.204	1.840	5.683	0.084
Godard *et al*. [16]		✓	0.544	10.94	11.76	0.193
Ours			0.383	5.321	10.47	0.478

Table 2. Results on the Make3D dataset [42]. Similar to ours, Godard *et al*. [16] do not utilize any of the Make3D data during training, and directly apply the model trained on KITTI+Cityscapes to the test set. Following the evaluation protocol of [16], the errors are only computed where depth is less than 70 meters in a central image crop.

Threshold: % of y_i s.t. $\max(\frac{y_i}{y_i^*}, \frac{y_i^*}{y_i}) = \delta < thr$

Abs Relative difference: $\frac{1}{|T|} \sum_{y \in T} |y - y^*| / y^*$

Squared Relative difference: $\frac{1}{|T|} \sum_{y \in T} ||y - y^*||^2 / y^*$

RMSE (linear): $\sqrt{\frac{1}{|T|} \sum_{y \in T} ||y_i - y_i^*||^2}$

RMSE (log): $\sqrt{\frac{1}{|T|} \sum_{y \in T} ||\log y_i - \log y_i^*||^2}$

Method	**Seq.** 09	**Seq.** 10
ORB-SLAM (full)	$\mathbf{0.014 \pm 0.008}$	$\mathbf{0.012 \pm 0.011}$
ORB-SLAM (short)	0.064 ± 0.141	0.064 ± 0.130
Mean Odom.	0.032 ± 0.026	0.028 ± 0.023
Ours	$\mathbf{0.021 \pm 0.017}$	$\mathbf{0.020 \pm 0.015}$

Table 3. Absolute Trajectory Error (ATE) on the KITTI odometry split averaged over all 5-frame snippets (lower is better). Our method outperforms baselines with the same input setting, but falls short of `ORB-SLAM (full)` that uses strictly more data.

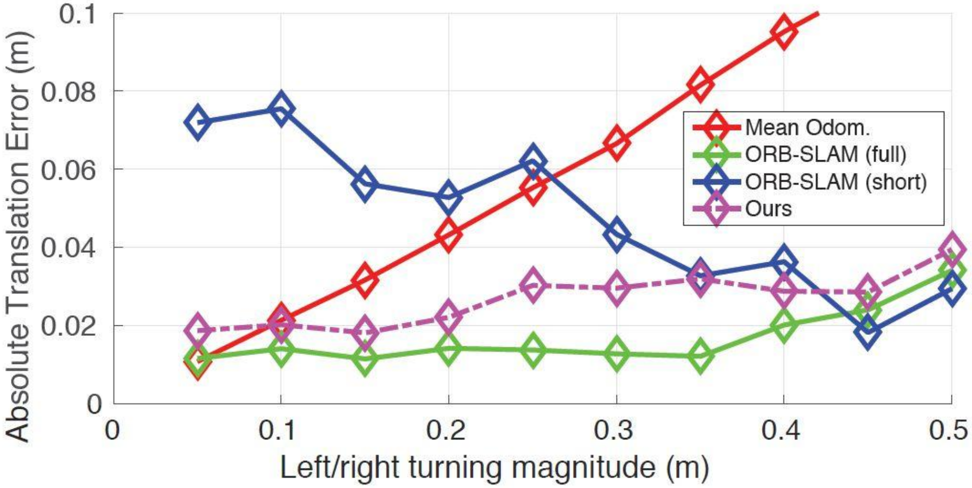


Figure 9. Absolute Trajectory Error (ATE) at different left/right turning magnitude (coordinate difference in the side-direction between the start and ending frame of a testing sequence). Our method performs significantly better than `ORB-SLAM (short)` when side rotation is small, and is comparable with `ORB-SLAM (full)` across the entire spectrum.

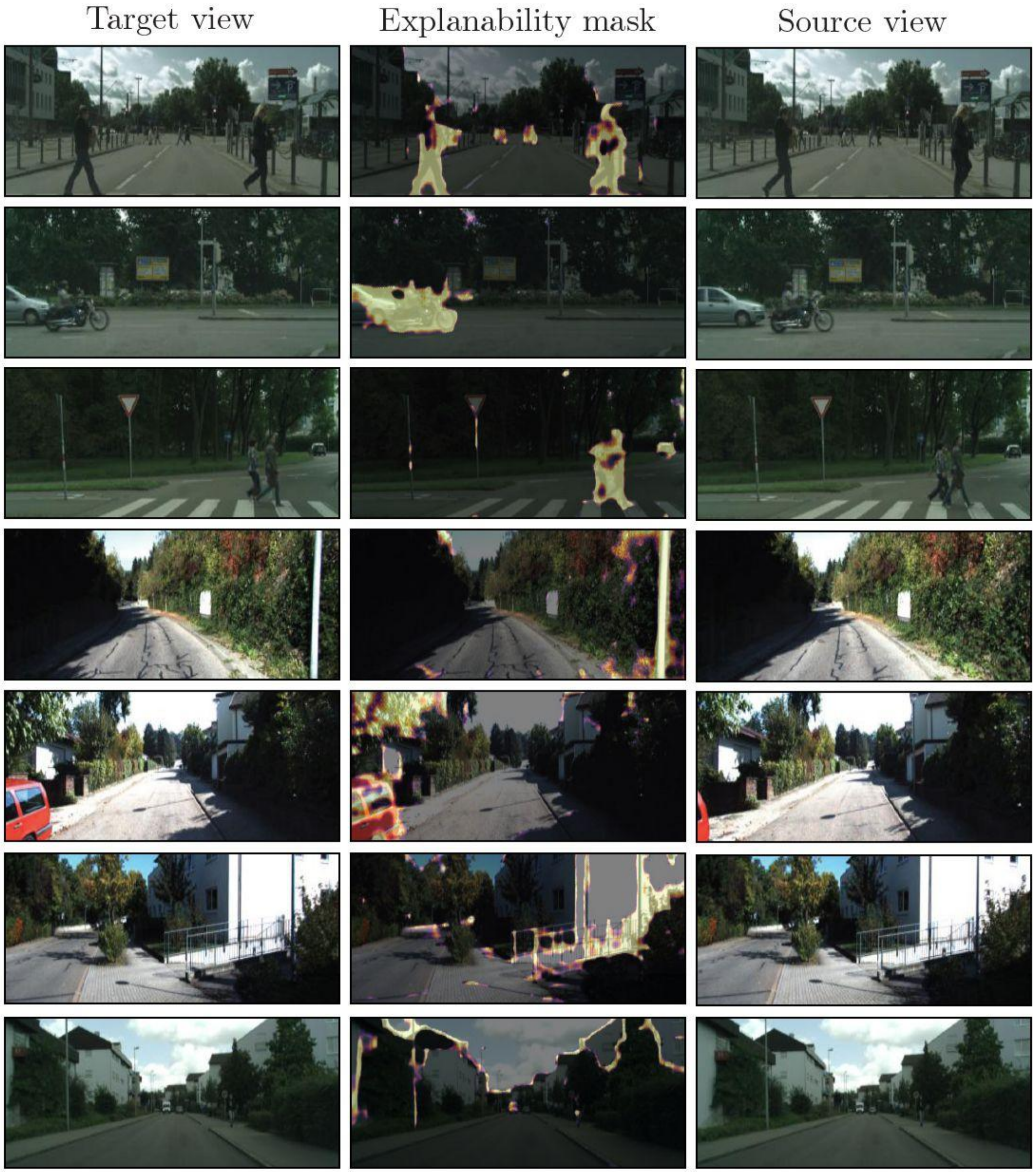


Figure 10. Sample visualizations of the explainability masks. Highlighted pixels are predicted to be unexplainable by the network due to motion (rows 1–3), occlusion/visibility (rows 4–5), or other factors (rows 7–8).

Conclusions

- Major challenges (not addressed):
  - Estimate scene dynamics and occlusions
  - Generalize for unknown camera types/calibrations
  - Learn full 3D volumetric representations
- Assumptions:
  - Pose network – uses image correspondences
  - Depth network – recognizes common structural features
- Extend to object detection and semantic segmentation