CAM-Convs: Camera-Aware Multi-Scale Convolutions for Single-View Depth

Jose M. Facil[1] **Benjamin Ummenhofer**[2,3] **Huizhong Zhou**[2] Luis Montesano[1,4] Thomas Brox*[2] Javier Civera*[1]

[1]University of Zaragoza [2]University of Freiburg [3]Intel Labs [4]Bitbrain * Equal Contribution

CVPR LONG BEACH CALIFORNIA June 16-20, 2019

Problem Statement

- **Overfitting**
  Overfitting a benchmark is common in computer vision. Datasets have strong bias that make researchers overconfident about their methods.
- **Camera Overfitting**
  We show that single-view depth models do not generalize to images from different cameras.
- **Our Proposal**
  Novel convolution that adresses the variability of the camera's internal parameters. Using different cameras without performance degradation.

Architecture and Training

- **U-Net Architecture**
  Skip-Conections between encoder and decoder
- **CAM-Convs Mapping from RGB to 3D**
  CAM-Convs on every encoder-decoder connection.
- **Intermediate Predictions**
  Pyramid-resolution predictions for improved internal features and faster convergence
- **Training Outputs**

ξ: Inverse depth: $\xi = \frac{1}{d}$

C: Depth confidence: $\hat{c}(i,j) = e^{-|\xi(i,j)-\hat{\xi}(i,j)|}$

$\mathbf{n}$: Surface Normals

- **Losses**

$\mathcal{L}_{total} = \mathcal{L}_d + \mathcal{L}_g + \mathcal{L}_c + \mathcal{L}_n$

-Depth:

$$\mathcal{L}_d = \sum_{i,j} \left|\xi(i,j) - \hat{\xi}(i,j)\right|.$$

-Confidence:

$$\mathcal{L}_c = \sum_{i,j} \left|c(i,j) - \hat{c}(i,j)\right|.$$

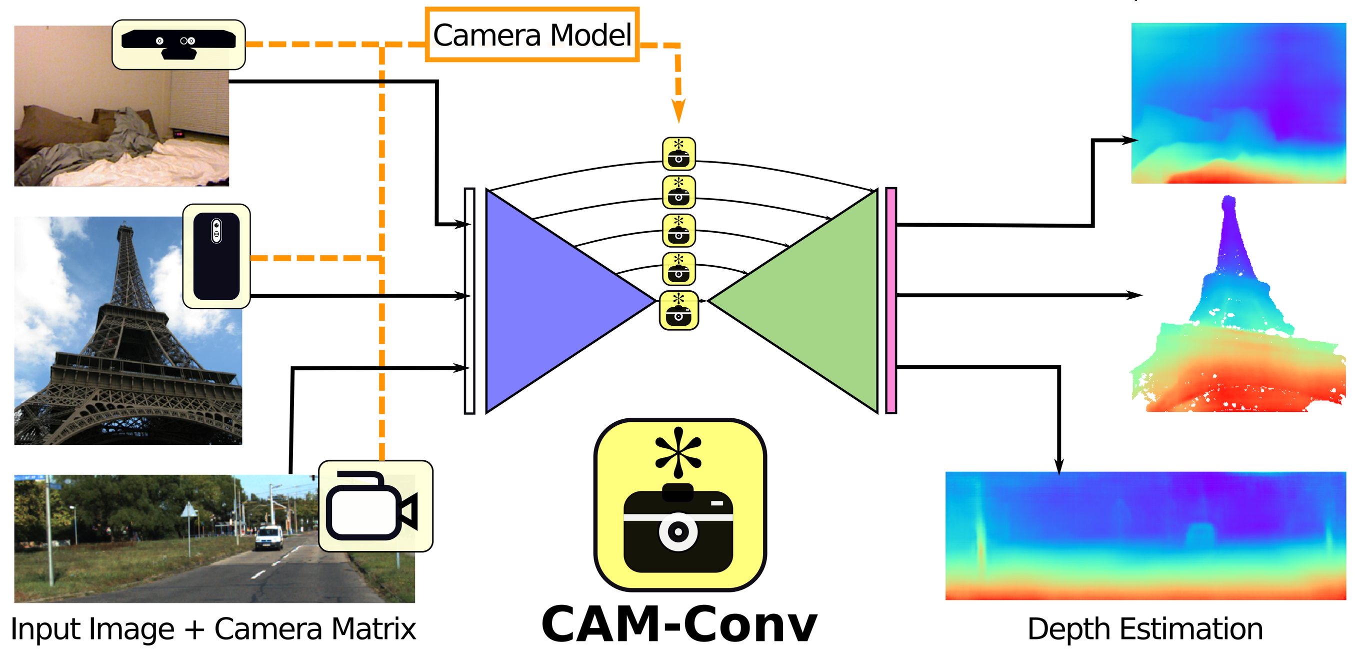
-Scale-Invariant Gradient:

$$\mathcal{L}_g = \sum_{h=\{1,2,4,8,16\}} \sum_{i,j} \left|\left|\mathbf{g}_h[\xi](i,j) - \mathbf{g}_h[\hat{\xi}](i,j)\right|\right|_2.$$

-Surface Normals:

$$\mathcal{L}_n = \sum_{i,j} \left|\left|\mathbf{n}(i,j) - \hat{\mathbf{n}}(i,j)|\right|\right|_2.$$

where $\mathbf{g}_h[d](i,j) = \left(\frac{d(i+h,j)-d(i,j)}{|d(i+h,j)+d(i,j)|}, \frac{d(i,j+h)-d(i,j)}{|d(i,j+h)+d(i,j)|}\right)^\top$

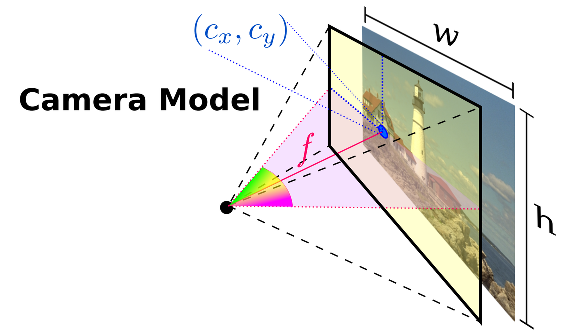


Acknowledgement

This project was in part funded by the Spanish government, the EU Horizon 2020 project Trimbot2020, the Aragón government and Fundación CAI-Ibercaja. We also thank Facebook for their P100 server donation and gift funding; and Nvidia for their Titan X and Xp donation.

Proposed Camera-Aware Convolution

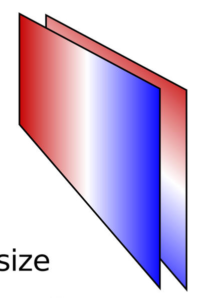
- **Camera Model by Convolution Channels Concatenation**
  Precomputed pixel-wise coordinates and field-of-view-maps are fed along with the input features to the convolution operation. **CAM-Convs** uses the idea behind Coord-Convs but incorporating the camera calibration.



-Centered Coordinates (cc):

$$cc_x = \begin{bmatrix} 0-c_x \\ 1-c_x \\ \vdots \\ w-c_x \end{bmatrix}_{w\times1} \cdot \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}_{h\times1}^\top \quad cc_y = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}_{w\times1} \cdot \begin{bmatrix} 0-c_y \\ 1-c_y \\ \vdots \\ h-c_y \end{bmatrix}_{h\times1}^\top$$

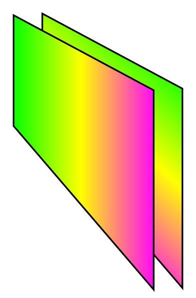
c_x, c_y: principal point f : focal length w, h : sensor size



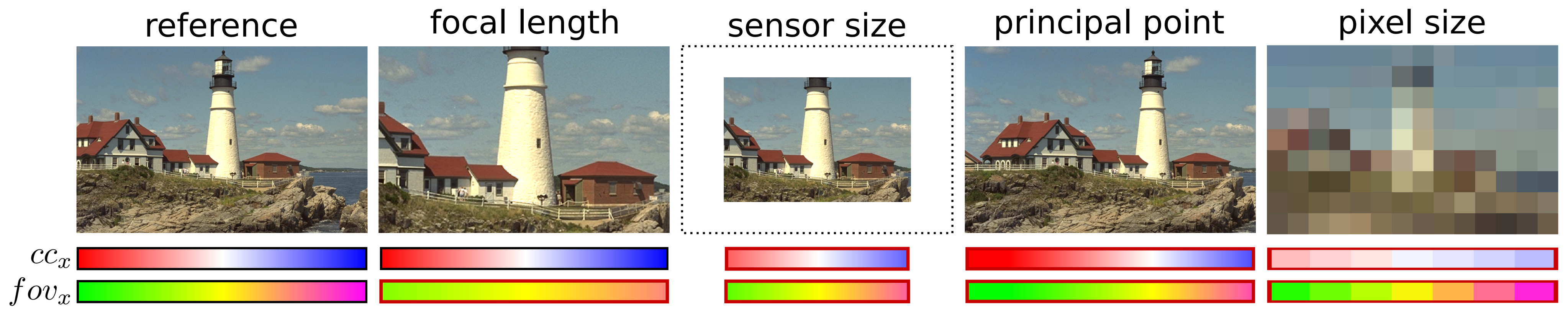
-Field of View Maps (fov):

$$fov_x[i,j] = \arctan\left(\frac{cc_x[i,j]}{f}\right)$$

$$fov_y[i,j] = \arctan\left(\frac{cc_y[i,j]}{f}\right)$$



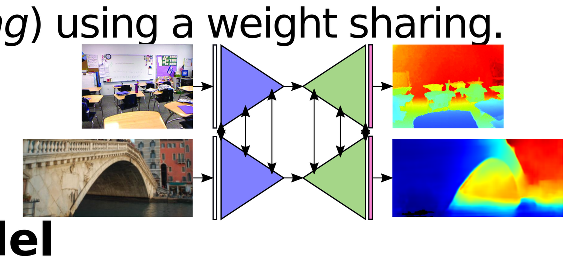
- **How changes on Camera Intrinsics Affect the CAM-Convs?**



Camera Generalization Analysis

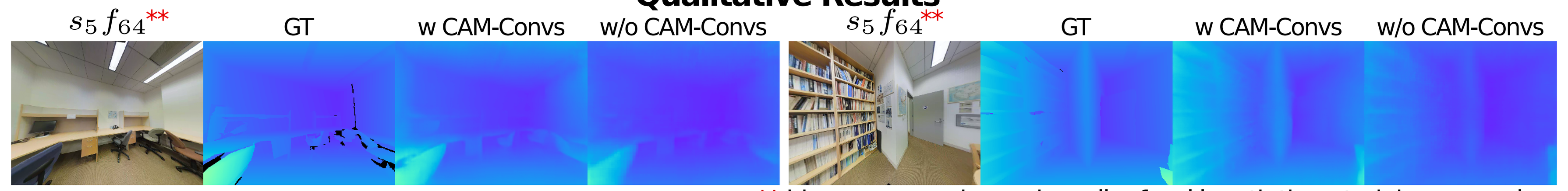
- **Multiple Cameras in the Same Domain**
  Goal: focus on the influence of the camera instead of different-domain.
  Experiments in de **2D-3D Semantics Dataset** (3-fold cross validation), generating images, from equirectangular images, with different camera intrinsics.
- **Comparison of Different *Naïve* Approaches**
  - To train and/or test with multiple focals using focal normalization.
  - To train and/or test with multiple sensor sizes using image resizing.
  - To train with many sensor sizes (if not *resizing*) using a weight sharing.

Focal Normalization: $\xi = \frac{f_n}{f}\tilde{\xi}$ Siamese Architecture:



- **Overfitting to a Single Camera Model**
  Train with one/multiple camera models and test on different ones. Compare with the ***same-camera baselines***. Standard networks overfit the camera they are trained with.
- **Robust Generalization with CAM-Convs**
  CAM-Convs can train with multiple cameras without compromising performance (beating ***same-camera baselines***) and generalize to unseen models.

Qualitative Results



** bigger sensor size and smaller focal length than training examples

Notation

$s_1 f_6$: set with sensor size s_1 and focal f_6

$s_1 s_4 f_5 f_6$: set with sensor size s_1 or s_4 and focal f_5 or f_6

Sensor Size	s_1	s_2	s_3	s_4	s_5
	256 × 192	192 × 256	224 × 224	128 × 96	320 × 320

Focal Length	f_{72}	f_{128}	f_{64}
	72	128	64

$f_{72} f_{128}$ means randomly sampling focal of 72 or 128

$\mathcal{U} f_{72} f_{128}$ means uniformly random sampling from 72 yo 128

Quantitative Results

Multiple Focal Length

Test set	Train set	l1.inv $1/m$	rmse m	sc.inv $lg(m)$
$s_1 f_{64}$	$s_1 f_{64}$*	0.184	0.378	0.0347
	$s_1 f_{72}$	0.193	0.395	0.0354
	$s_1 f_{128}$	0.318	0.572	0.0483
	$s_1 f_{72} f_{128}$*	0.659	0.864	0.0614
	$s_1 f_{72} f_{128}$	0.189	0.387	0.0361
$s_1 f_{72}$	$s_1 f_{72}$*	0.17	0.4	0.0354
	$s_1 f_{128}$	0.272	0.564	0.0459
	$s_1 f_{72} f_{128}$*	0.552	0.888	0.0609
	$s_1 f_{72} f_{128}$	0.175	0.404	0.0364
$s_1 f_{128}$	$s_1 f_{128}$*	0.141	0.51	0.0387
	$s_1 f_{72}$	0.133	0.524	0.0411
	$s_1 f_{72} f_{128}$*	0.208	0.813	0.063
	$s_1 f_{72} f_{128}$	**0.132**	**0.504**	**0.038**

smallest the best

Multiple Camera

Test	Train	l1.inv $1/m$	rmse m	sc.inv $lg(m)$
$s_1 \mathcal{U} f_{72} f_{128}$	$s_1 \mathcal{U} f_{72} f_{128}$	0.15	0.46	0.037
	$s_2 \mathcal{U} f_{72} f_{128}$	0.175	0.51	0.0422
	$s_1 s_2 \mathcal{U} f_{72} f_{128}$	0.153	0.484	0.0401
	$s_1\ s_2\ s_3 \mathcal{U} f_{72} f_{128}$†	0.179	0.742	0.064
	CAM-Convs‡	**0.144**	**0.433**	**0.0312**
$s_2 \mathcal{U} f_{72} f_{128}$	$s_1 \mathcal{U} f_{72} f_{128}$	0.151	0.44	0.038
	$s_2 \mathcal{U} f_{72} f_{128}$	0.133	0.412	0.0323
	$s_1 s_2 \mathcal{U} f_{72} f_{128}$	0.139	0.436	0.0352
	$s_1\ s_2\ s_3 \mathcal{U} f_{72} f_{128}$†	0.16	0.622	0.0514
	CAM-Convs‡	**0.131**	**0.39**	**0.0265**
$s_3 \mathcal{U} f_{72} f_{128}$	$s_3 \mathcal{U} f_{72} f_{128}$	0.14	0.425	0.0336
	$s_1 \mathcal{U} f_{72} f_{128}$	0.143	0.44	0.0357
	$s_2 \mathcal{U} f_{72} f_{128}$	0.145	0.435	0.0356
	$s_1 s_2 \mathcal{U} f_{72} f_{128}$	0.143	0.451	0.0365
	CAM-Convs‡	**0.134**	**0.402**	**0.0283**
$s_5 f_{64}$	$s_5 f_{64}$	**0.227**	0.309	**0.0356**
	$s_1 f_{64}$	0.292	0.337	0.0598
	$s_1 s_2 \mathcal{U} f_{72} f_{128}$	0.369	0.44	0.0427
	CAM-Convs‡	0.236	**0.289**	0.0362

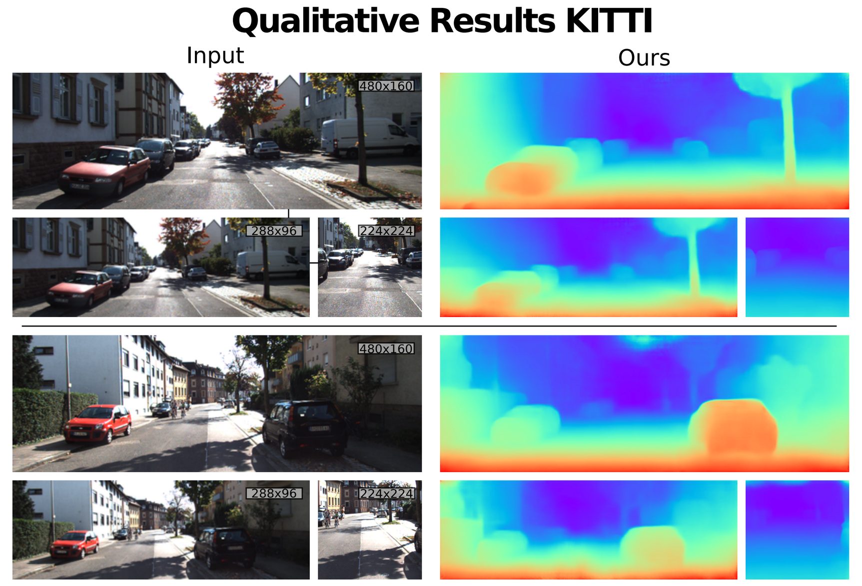
smallest the best

* trained without focal length normalization.
† images resized to s_1 during training.
‡ trained with in sensor sizes s_1, s_2 and $\mathcal{U} f_{72} f_{128}$.

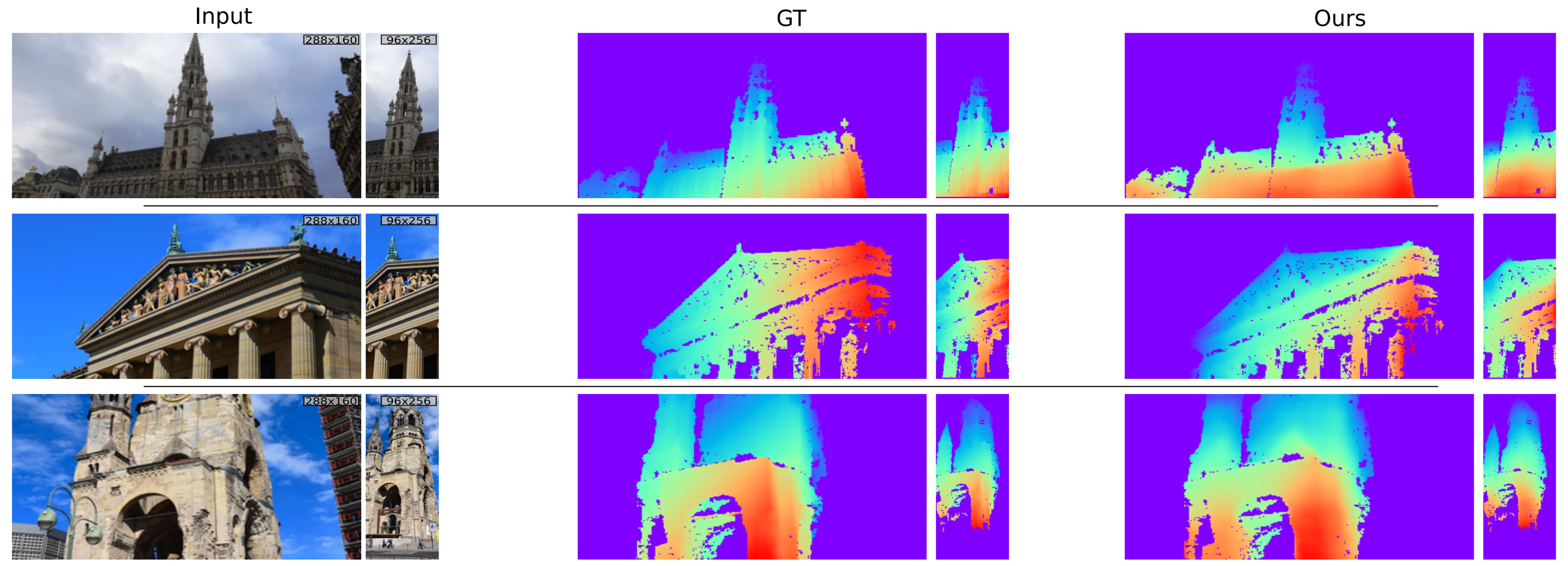
Experiments on Multiple Datasets

- **Train Multiple Datasets**
  - **Single Model** for Multiple Datasets
  - Train our Network on
    - KITTI
    - ScanNet
    - MegaDepth
    - Sun3D.
  - Camera data augmentation.
    - Focal length
    - Principal point shifting
    - Sensor size

Qualitative Results KITTI

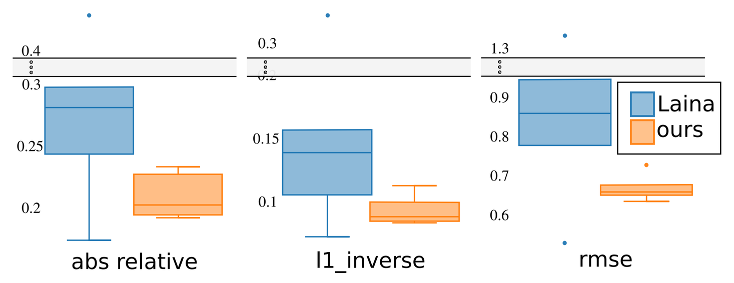


Qualitative Results MegaDepth

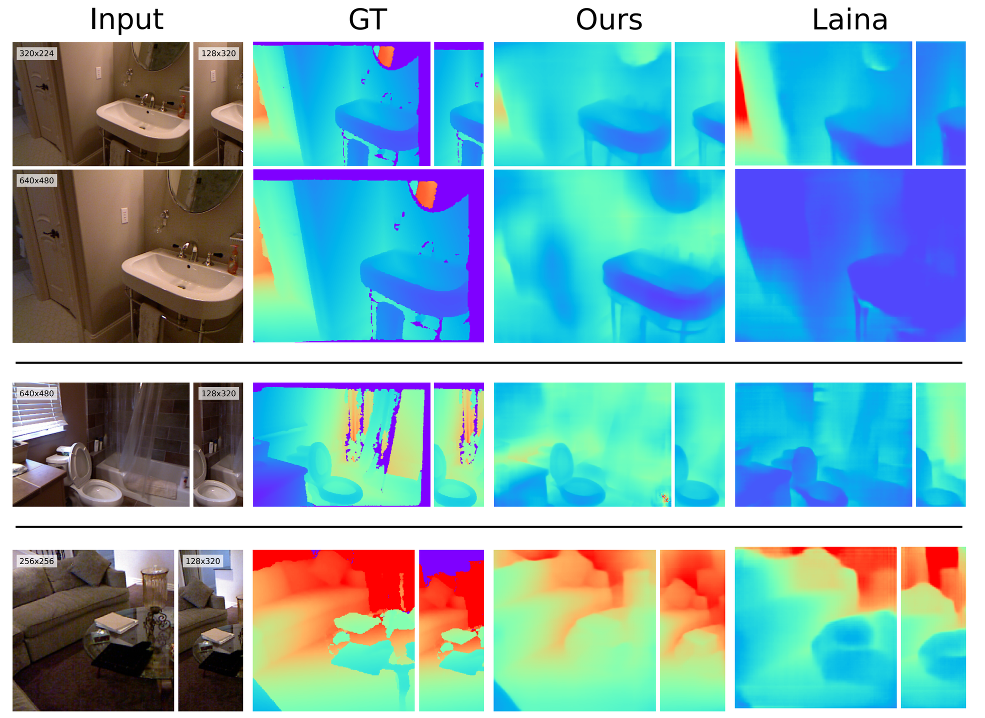


- **Testing Dataset**
  Test on NYU and compare with *Laina et al 3DV 2016*, trained exclusively on NYU. We test simulating 5 more extra cameras on the same test split in order to measure generalization across cameras.

Error Distribution NYU



Qualitative Results NYU



Project Page

- **Paper**
- **Code (coming soon)**
- **Poster**
- **Contact**