Codeway
Case Study

Guided Super-Resolution as Pixel-to-Pixel Transformation

ICCV 2019

Riccardo de Lutio | Stefano D'Aronco | Jan Dirk Wegner | Konrad Schindler

EcoVision Lab, Photogrammetry and Remote Sensing, ETH Zürich

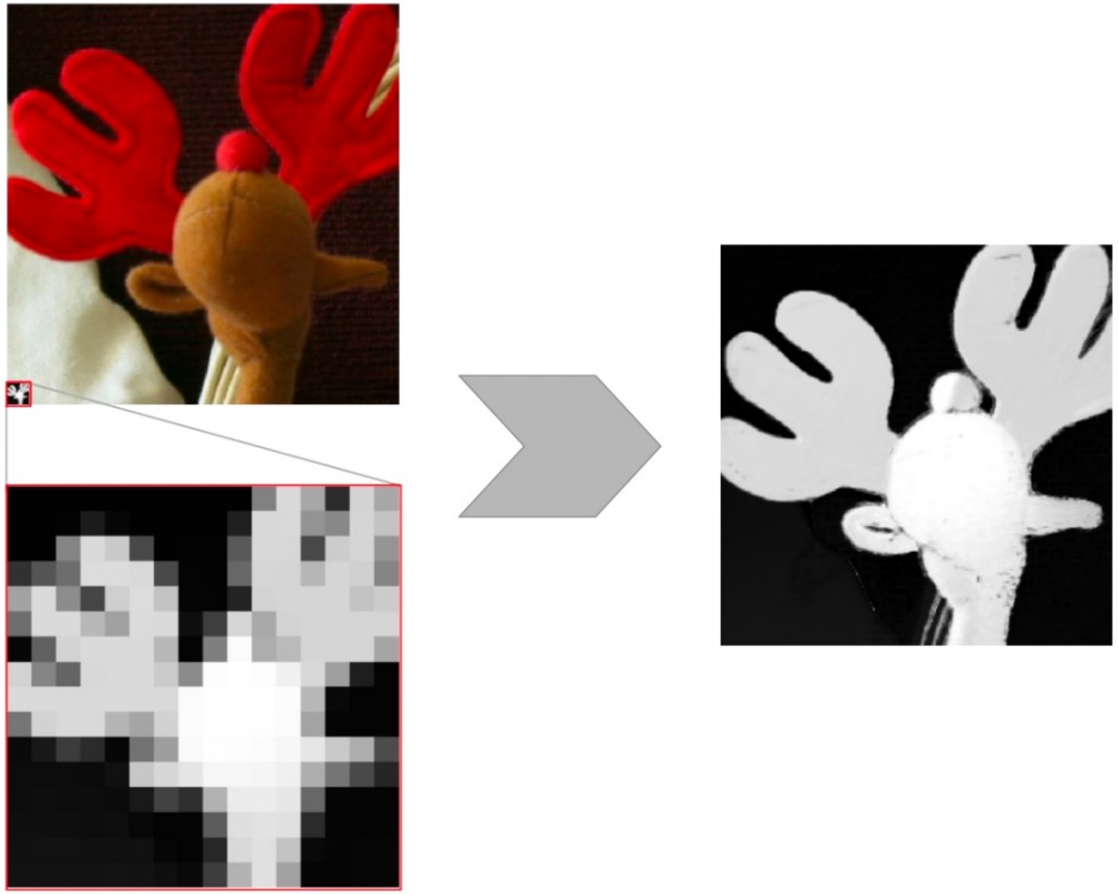
Halil Çağrı Bilgi

Outline

- Problem Definition
- Paper Overview
- Implementation Details
- Results of Test Data provided by the Authors
- Comparison
- Results of Codeway Data
- Discussion

Guided Super Resolution

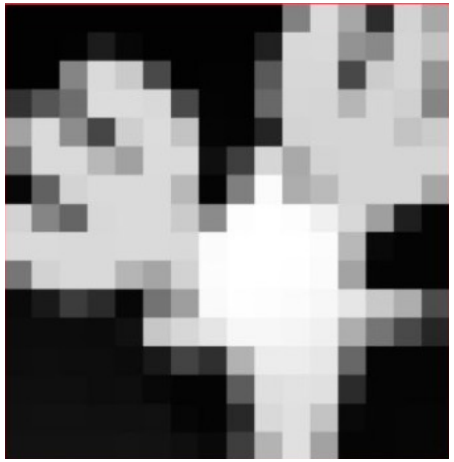
Aim
Given a low-resolution depth map and a high-resolution RGB guide image, the method should predict a high-resolution depth map.



How to solve?

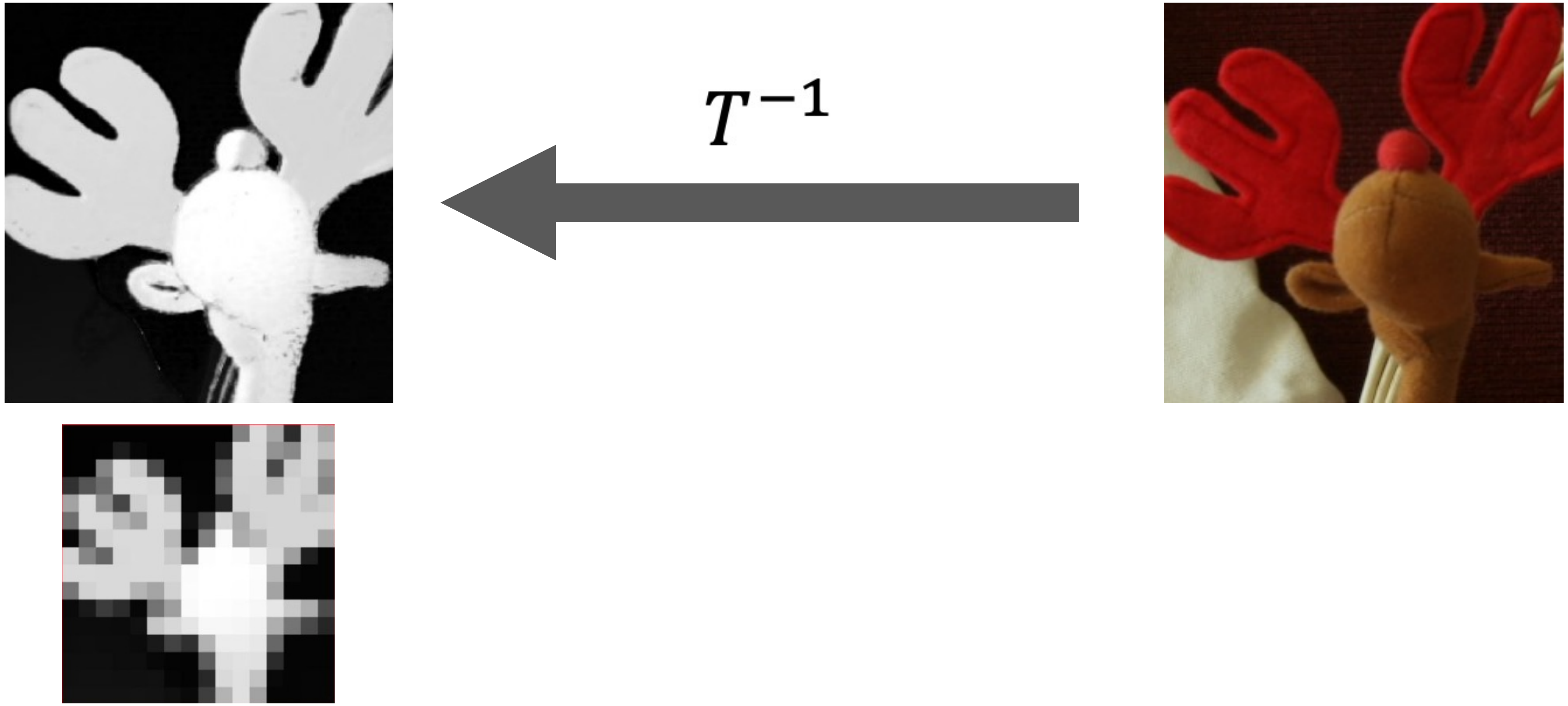
- Think this problem as an Inverse Problem

T^{-1}

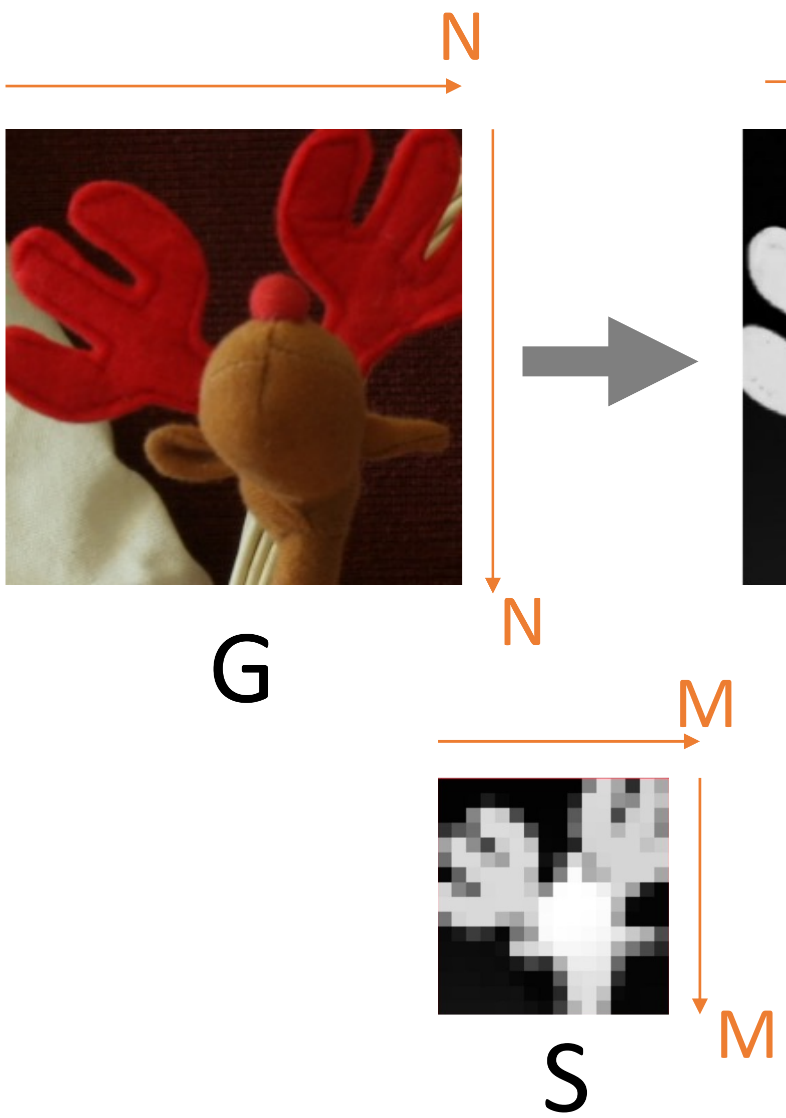


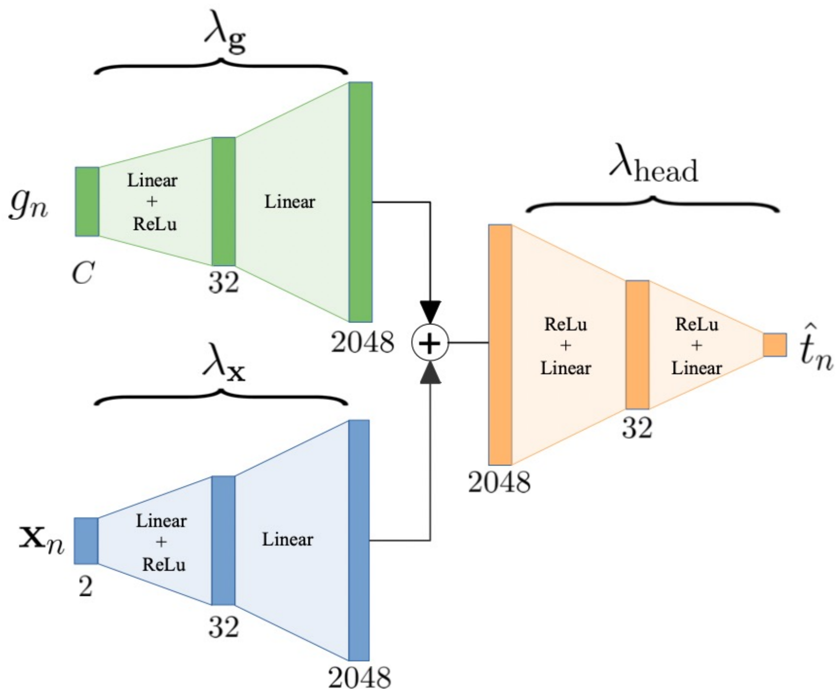
How to solve?

- Formulize as a pixel-wise mapping from one image domain to another



Proposed Solution





- Reformulate the problem as trying to find a function $f_\theta: R^{C+2} \rightarrow R$ with parameters θ that maps every guide pixel to a target pixel

How to train?

Objective $$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{m} \left| s_m - \langle f_{\boldsymbol{\theta}}(\mathbf{g}_n, \mathbf{x}_n) \rangle_{\mathsf{b}(m)} \right| + \lambda \|\boldsymbol{\theta}\|^2 . \quad (4)$$

1	2
2	3

Source

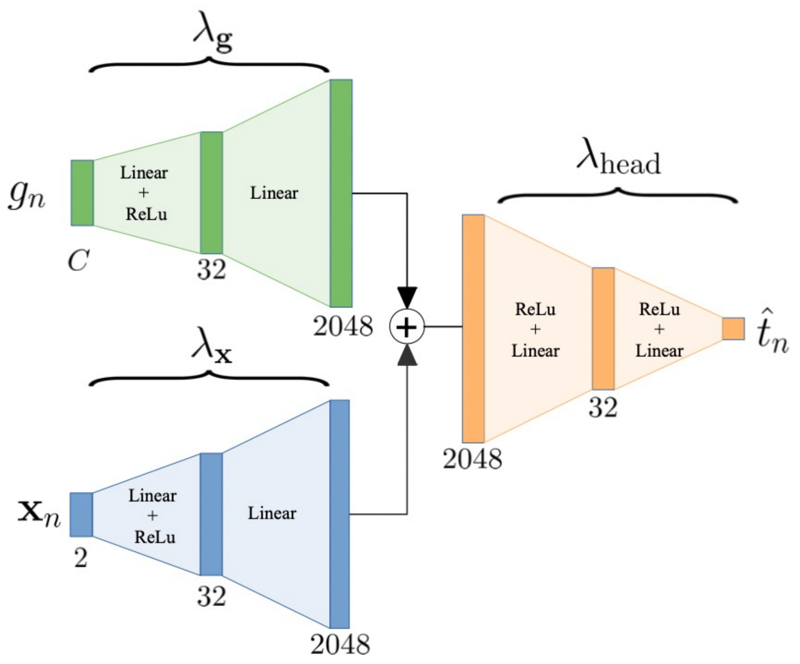
1	2	1	2	2	3	4	1
2	2	3	4	2	2	3	4
1	2	3	3	3	3	4	1
2	3	4	3	2	2	3	4
1	3	3	4	3	4	4	2
4	4	3	3	1	3	4	2
4	3	4	3	2	3	4	2
3	1	4	3	4	1	2	3

Target Prediction

- For a given S, T and G a perfect solution can always be found by choosing a sufficiently complex function f_θ
- To ensure the problem is solvable
    - Restrict f_θ to a function with reasonably low complexity.
    - Use l-2 penalty on the function weights

Inference

- Fit an individual set of weights using all pixels in Guide image as ”training data” and the low-resolution source as ”supervision, then make prediction (forward pass)



Experiment Settings

- Since we think every pixel as an independent training data, they train the model in batches of size 32 low-resolution pixels/blocks.

- Adam with lr=0.001

- 32.000 iteration steps

Quantitative Error Metrics

- Mean Absolute Error

$$\text{MAE} = \frac{1}{N^2}\sum_n |t'_n - t_n|$$

- Mean Square Error

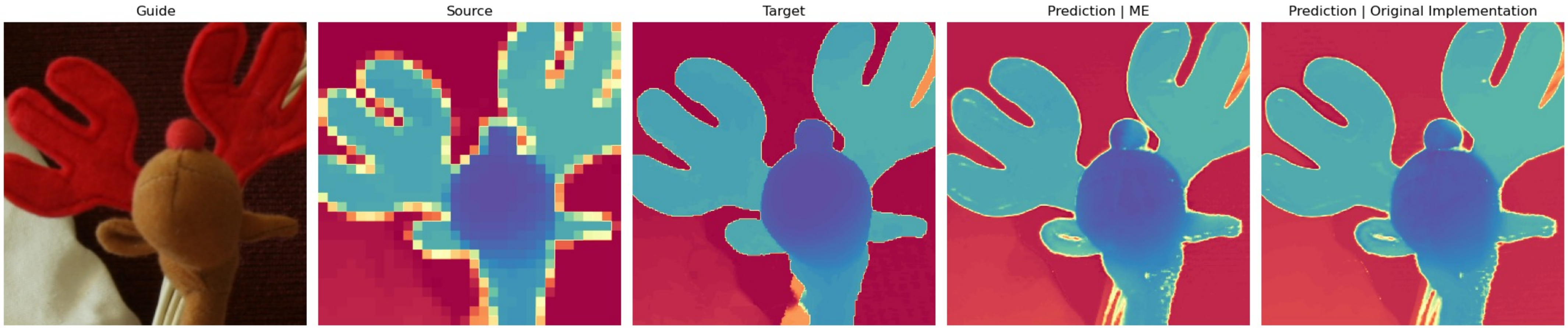
$$\text{MSE} = \frac{1}{N^2}\sum_n (t'_n - t_n)^2$$

- Percentage of Bad Pixels

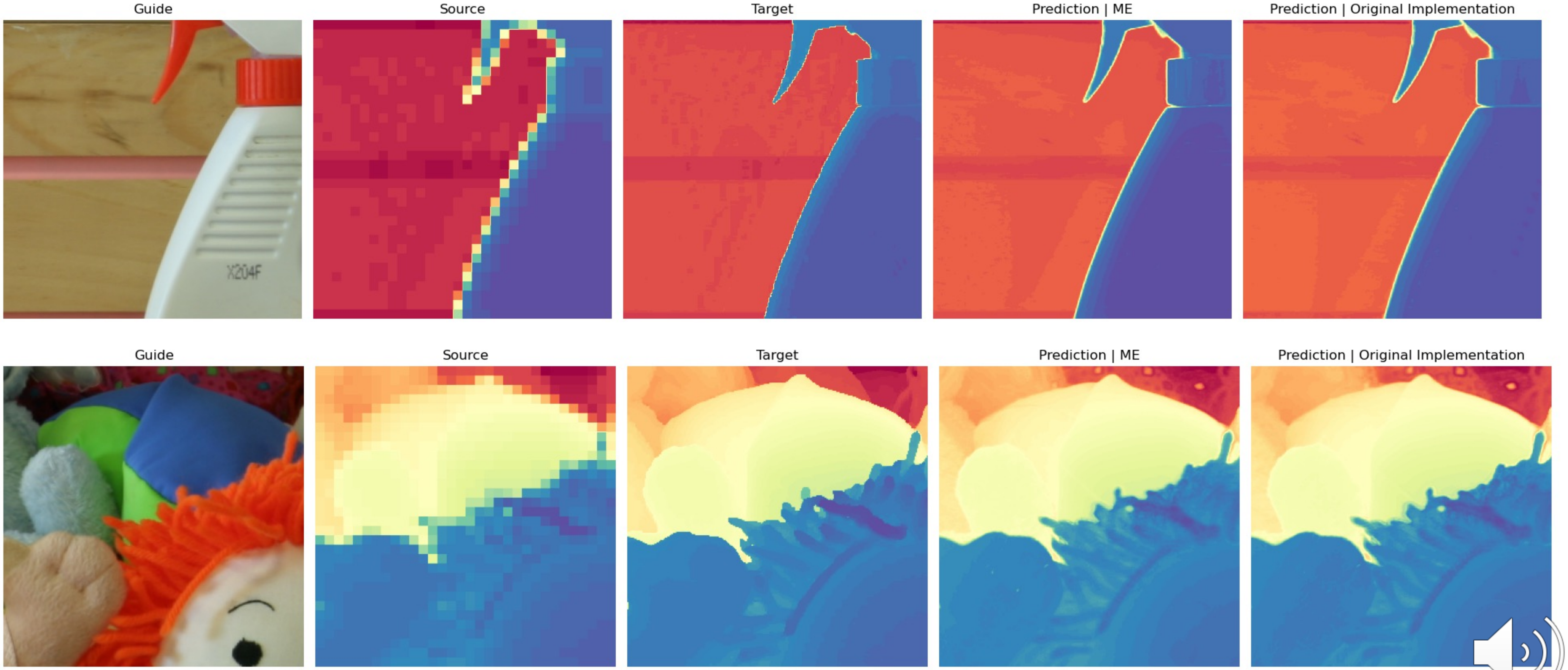
$$\text{PBP} = \frac{1}{N^2}\sum_n [|t'_n - t_n| > \delta] \quad \text{where } [.] = \begin{cases} 1 & if\ condition\ is\ True \\ 0 & if\ condition\ is\ False \end{cases}$$

Results on Test Data Provided by Authors

	My Implementation			Original Implementation		
	MAE	MSE	PBP	MAE	MSE	PBP
Example 1	2.200	65.145	0.228	1.949	64.710	0.159
Example 2	0.420	2.286	0.038	0.434	3.015	0.043
Example 3	0.786	3.494	0.176	0.876	3.654	0.213

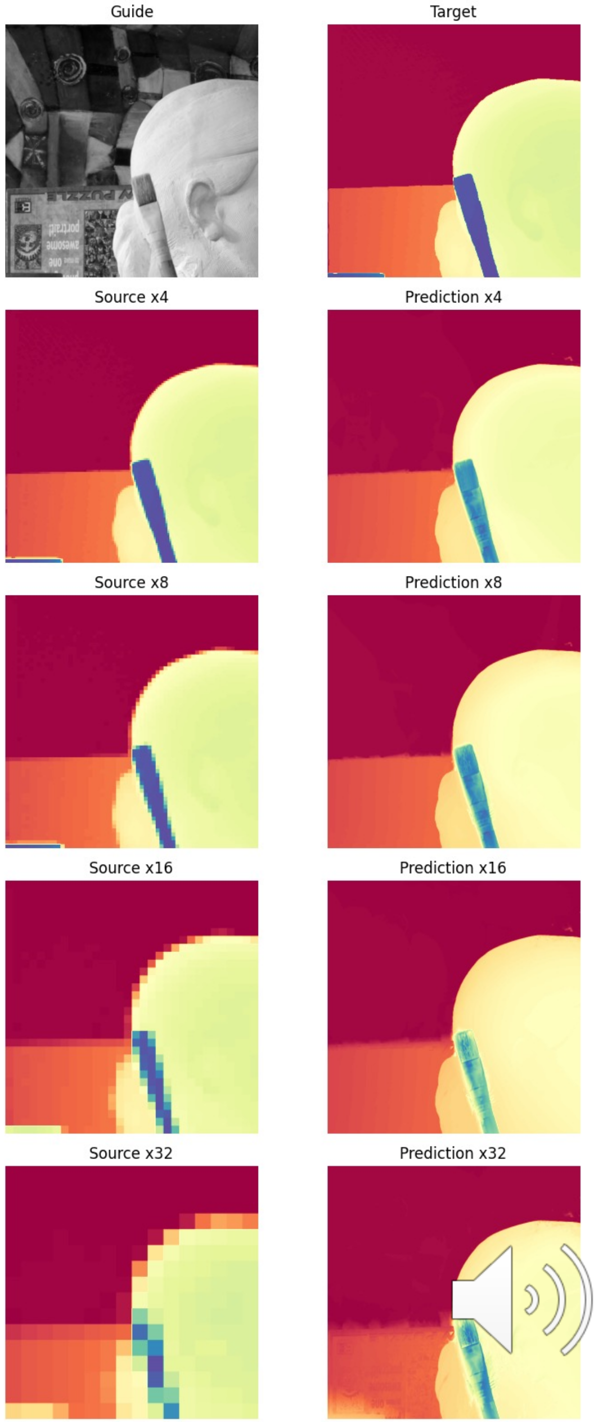


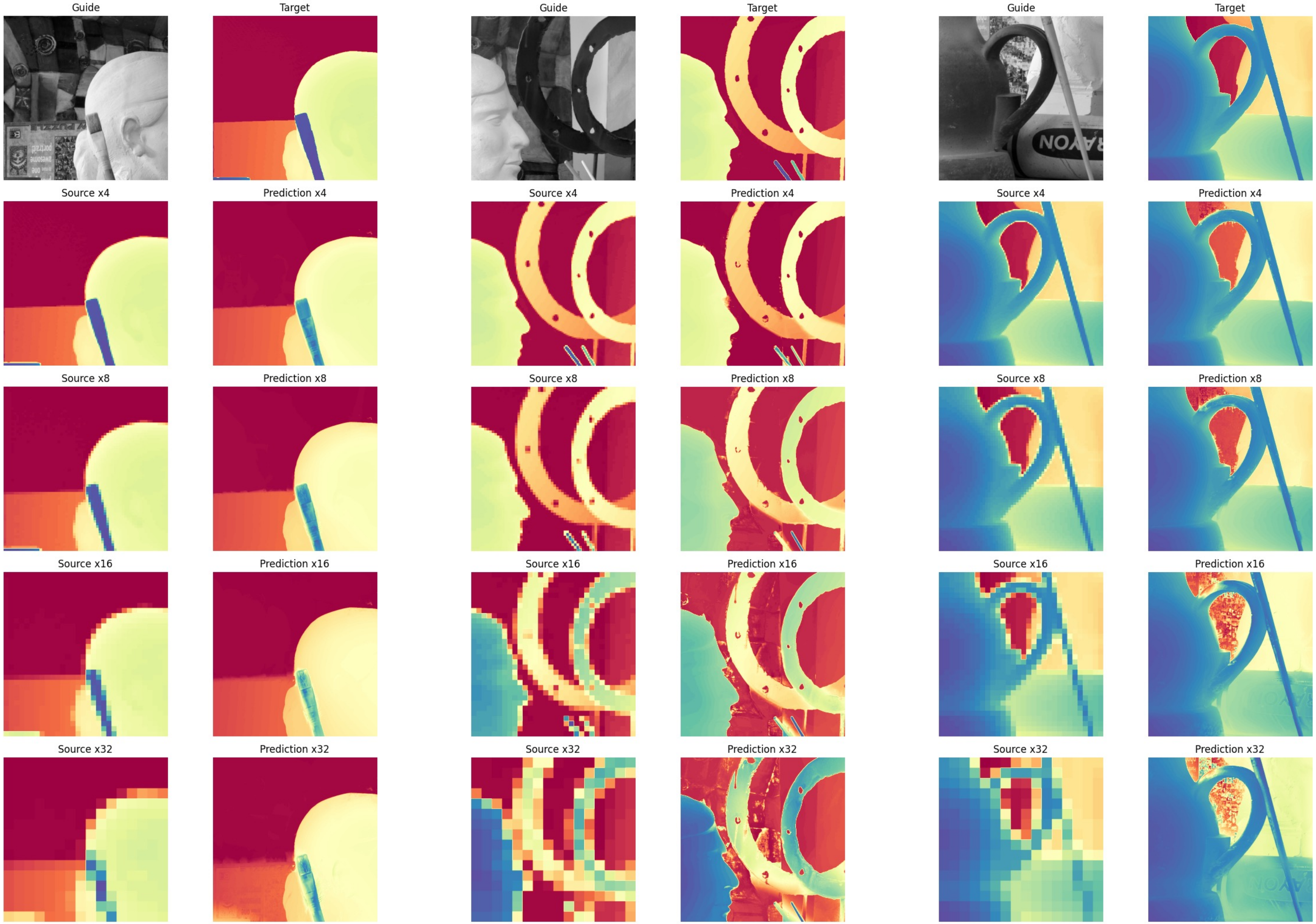
Qualitative Results

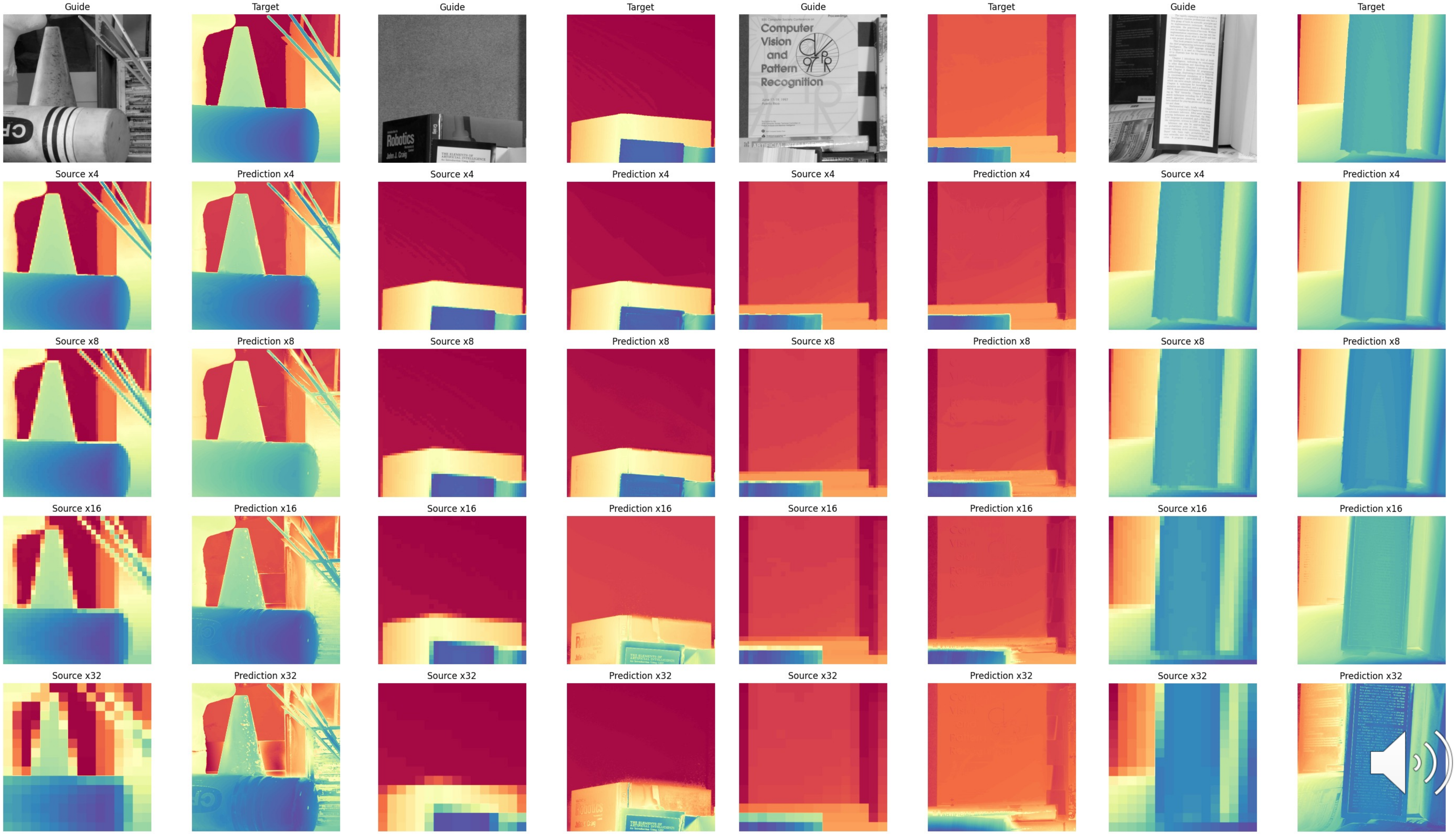


Results on Codeway Data

	Example 1			Example 2			Example 3		
	MAE	MSE	PBP	MAE	MSE	PBP	MAE	MSE	PBP
x4	0.896	33.477	0.089	1.066	29.313	0.096	0.956	14.592	0.120
x8	1.029	35.755	0.093	1.396	33.108	0.143	1.076	15.962	0.169
x16	1.264	38.396	0.159	2.342	50.252	0.354	3.859	55.624	0.349
x32	1.833	32.674	0.261	4.125	72.462	0.699	4.062	96.991	0.539




Guide
Target
Guide
Target
Guide
Target
Source x4
Prediction x4
Source x4
Prediction x4
Source x4
Prediction x4
Source x8
Prediction x8
Source x8
Prediction x8
Source x8
Prediction x8
Source x16
Prediction x16
Source x16
Prediction x16
Source x16
Prediction x16
Source x32
Prediction x32
Source x32
Prediction x32
Source x32
Prediction x32
NOYAR


Guide
Target
Guide
Target
Guide
Target
Guide
Target
Computer Vision and Pattern Recognition
Robotics
John J. Craig
THE ELEMENTS OF ARTIFICIAL INTELLIGENCE
Source x4
Prediction x4
Source x4
Prediction x4
Source x4
Prediction x4
Source x4
Prediction x4
Source x8
Prediction x8
Source x8
Prediction x8
Source x8
Prediction x8
Source x8
Prediction x8
Source x16
Prediction x16
Source x16
Prediction x16
Source x16
Prediction x16
Source x16
Prediction x16
Source x32
Prediction x32
Source x32
Prediction x32
Source x32
Prediction x32
Source x32
Prediction x32

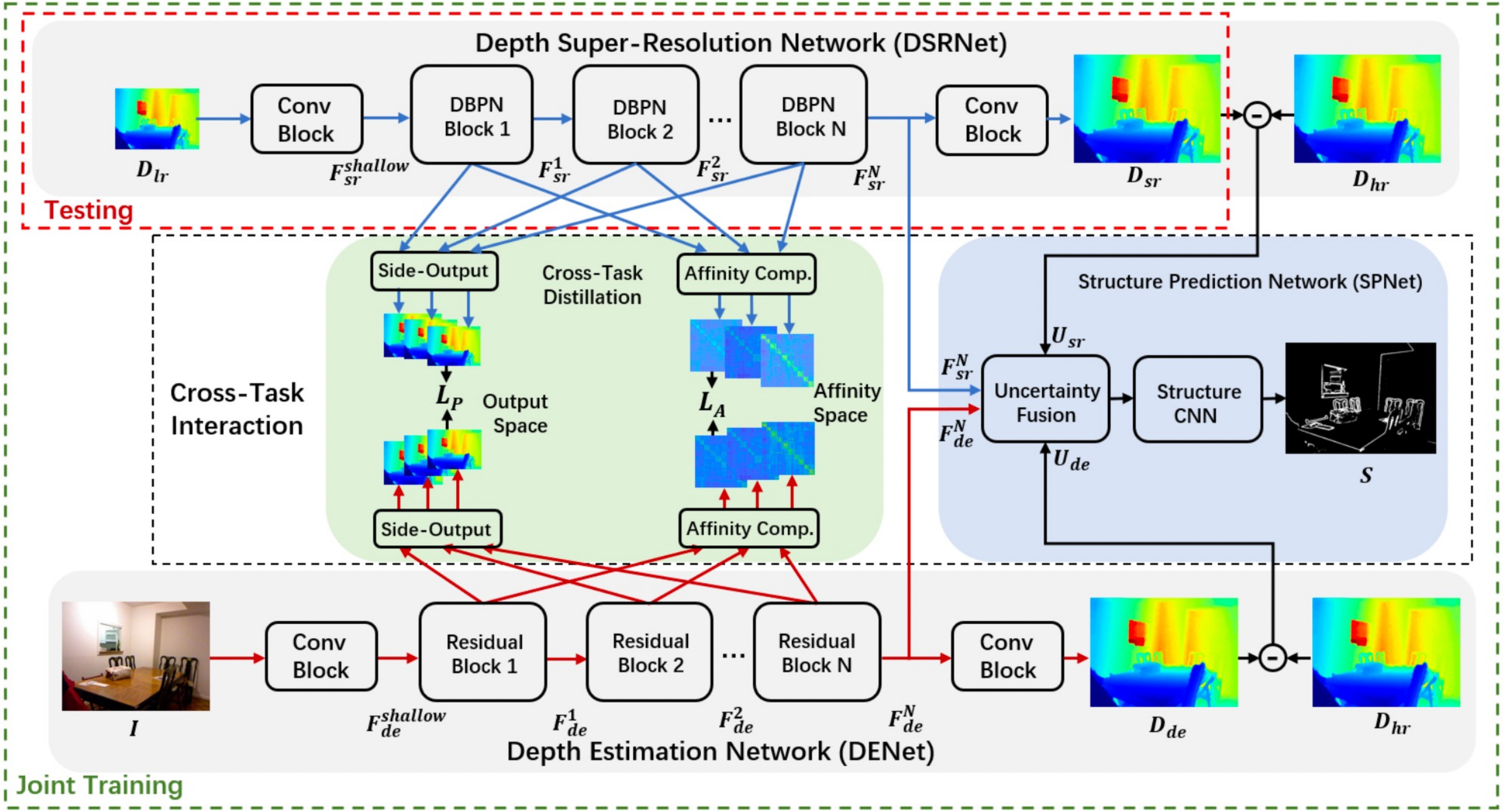
Limitation of the method

- Inference on one image requires to find optimal paramaters for that image, takes approximately 2 mins with x8 scaling. However, it can be applied to other datasets, who contains RGB and LR Depth images and requires to find the High-Res Depth map.

- Method is local, no global optimization is considered

- RGB image and depth map are captured by separate depth and RGB sensors with different resolutions and views, thus needing accurate calibration and rectification between them to obtain the registered pairs.

How to Improve?



Thanks.