



# Depth and Stereo

## What is a stereo camera?

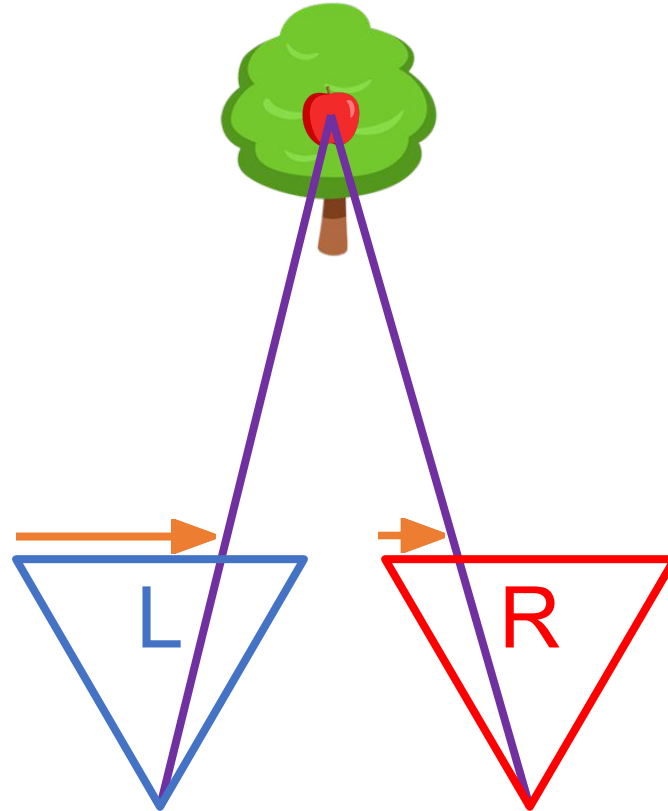
- Two cameras separated by a baseline





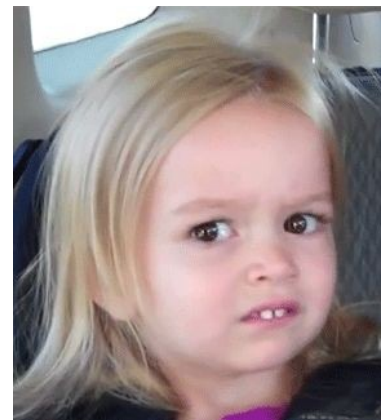
How do we get depth from stereo?

$$\textit{Disparity} \propto \frac{1}{\textit{Depth}}$$



Great, let's train a neural network with stereo pairs then!

How do we do this?



The trick is to pose the problem as an ***image reconstruction*** one.

If the network can predict the **right** image from the **left** one it means it has an internal understanding of depth.

We learn only from the data, through a **proxy** loss.

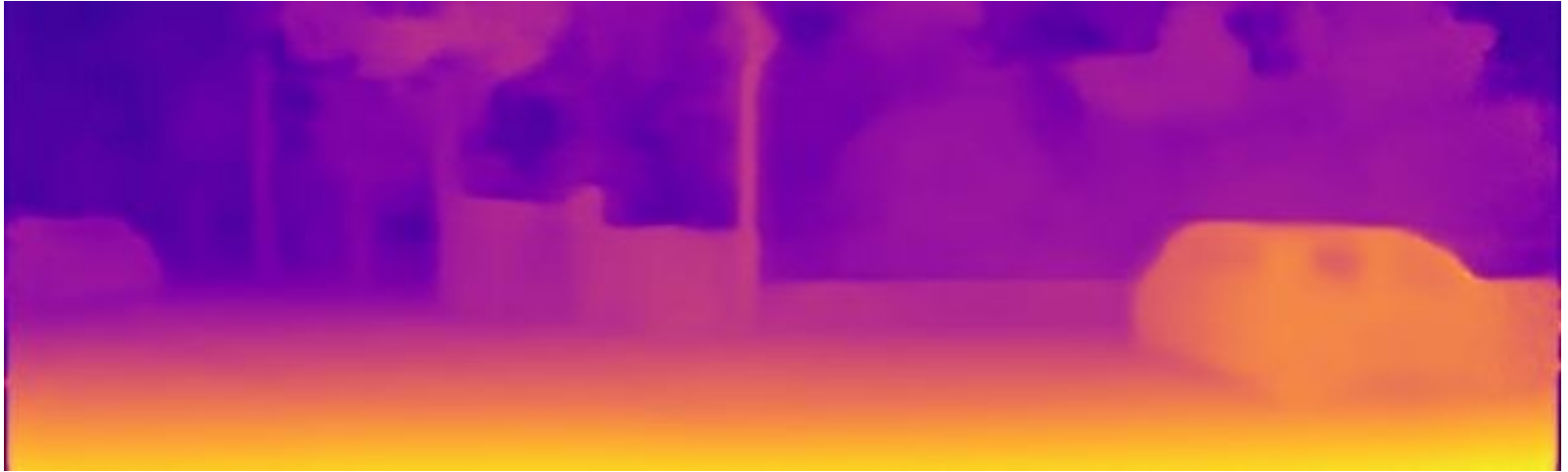
Hence we talk about **self-supervision**.

How to go from this?



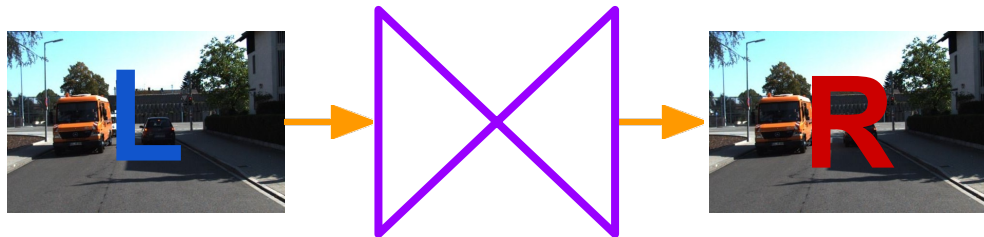


to this





Let's design a (naive) model.



It works! However:

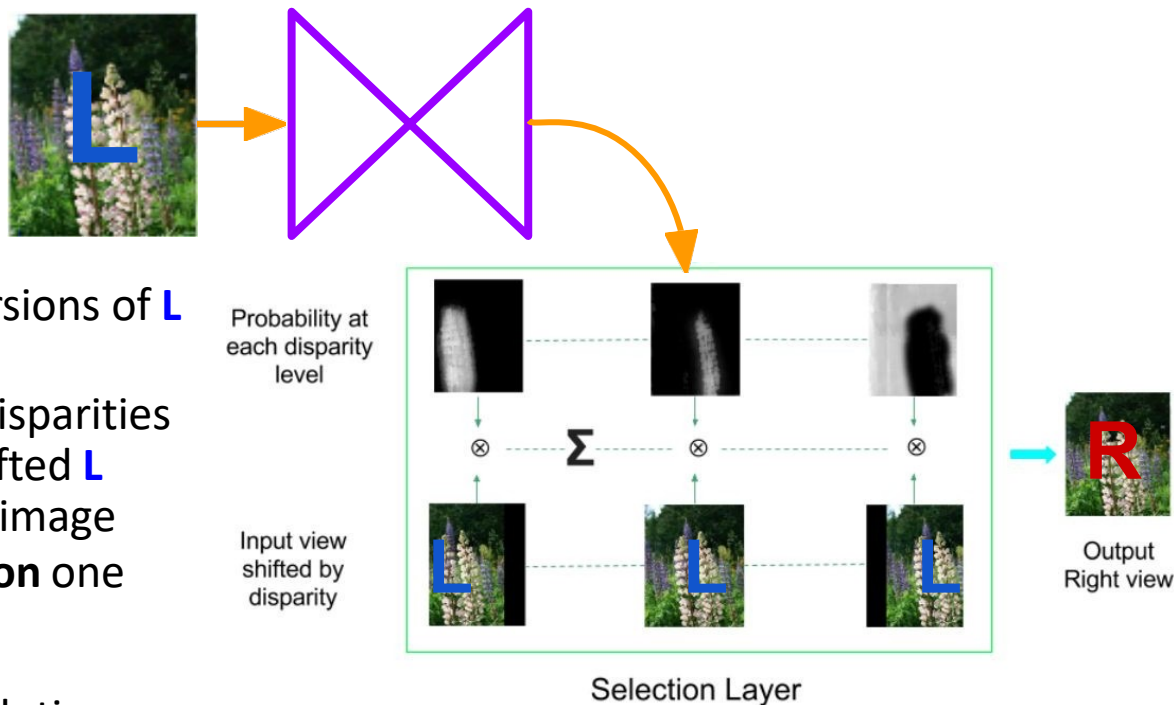
- Depth perception is *latent*
- We have no way to extract it

We need an **interpretable** internal representation.  
In stereo the obvious choice is **disparity**.

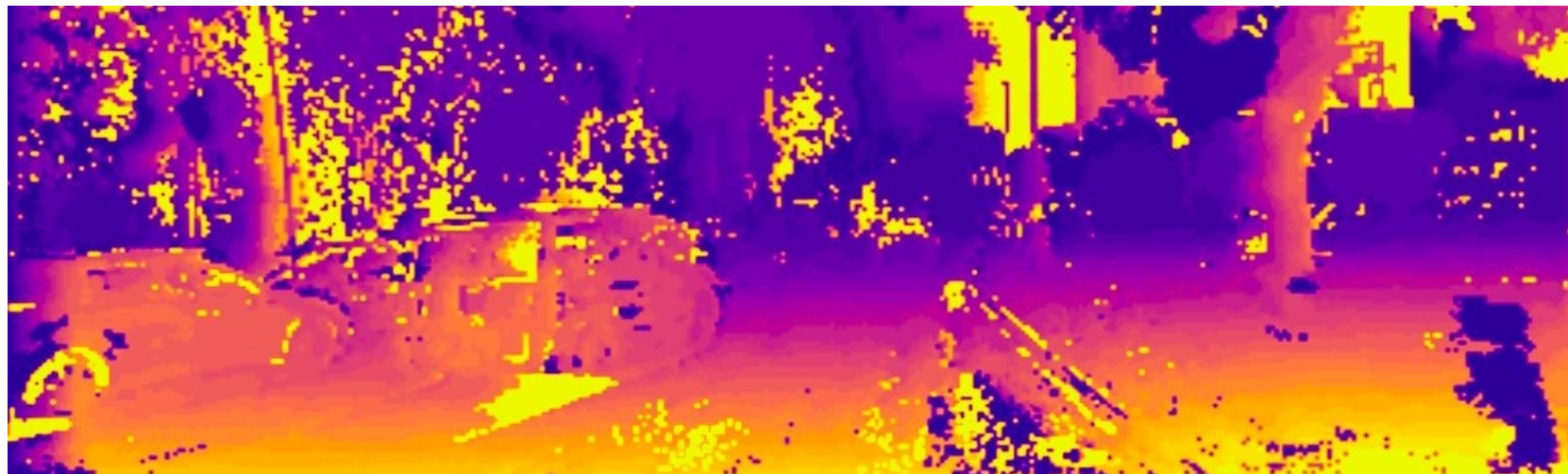
## Deep3D at ECCV2016 [1]

- Generate **R** from **N** shifted versions of **L**
- Each pixel predicts a discrete probability distribution over disparities
- Weights are used to blend shifted **L** images into one composite **R** image
- Loss is an **image reconstruction** one

- **N** becomes large for high resolutions which results in high memory usage
- No single disparity value predicted







Why can't we just **sample** the image?

# Geometry to the Rescue [2]

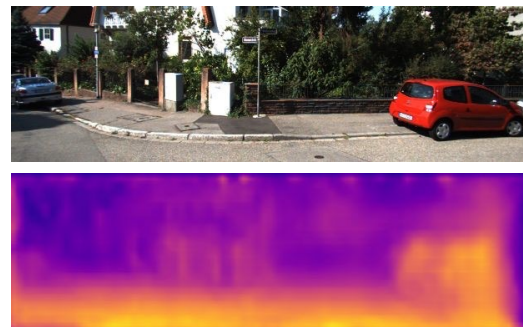
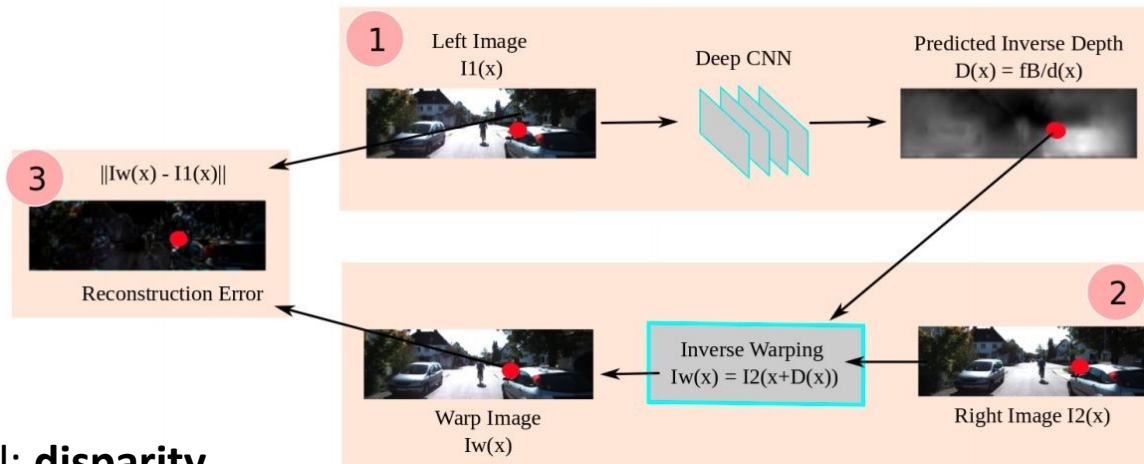
ECCV 2016

Idea:

- Warp **R** to generate **L**
- Single predicted value per pixel: **disparity**
- Multi-stage shallow decoder with coarse to fine prediction

Is warping differentiable?

- Not quite, so an approximation was used for the gradients
- Harder to optimize, training in stages





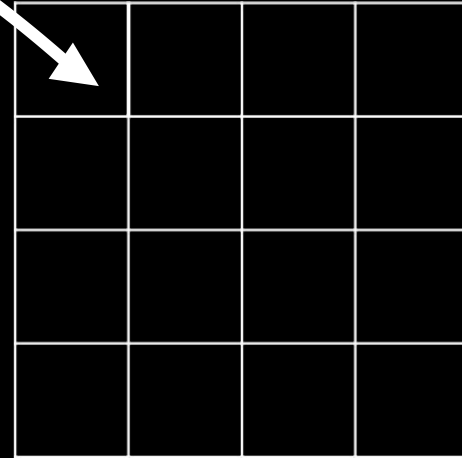
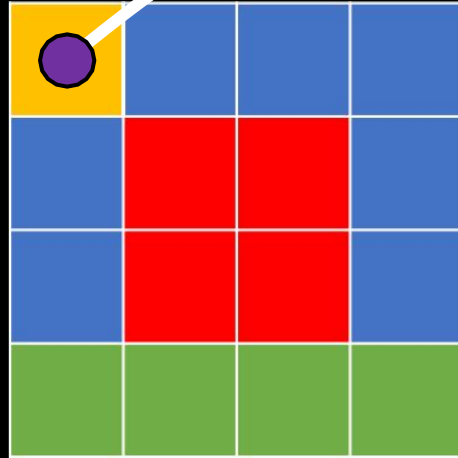
Can we make warping differentiable?



Forward mapping: where do source pixels go?

0	0	0	0
0	1	1	1
0	1	1	1
0	0	0	0

disparity

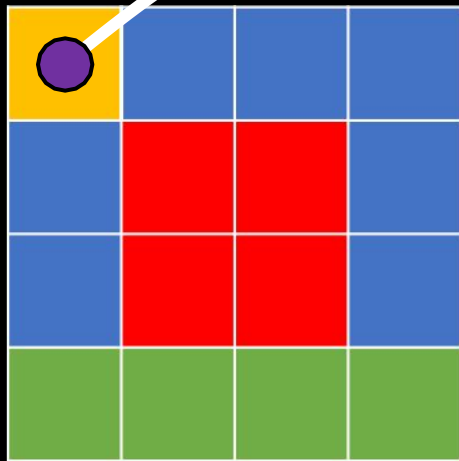


target

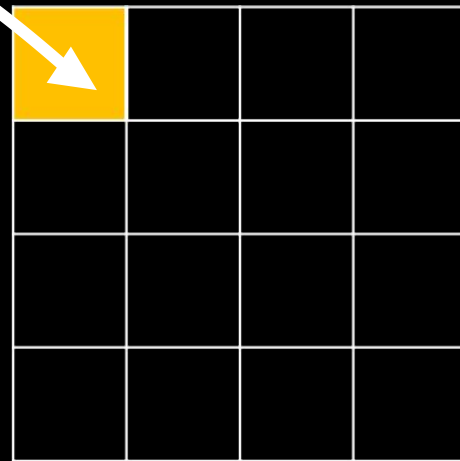
Forward mapping: where do source pixels go?

0	0	0	0
0	1	1	1
0	1	1	1
0	0	0	0

disparity



source

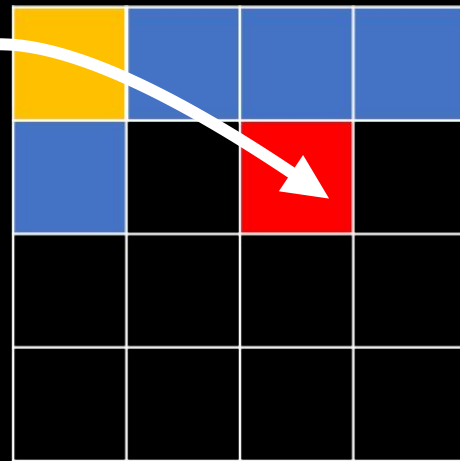
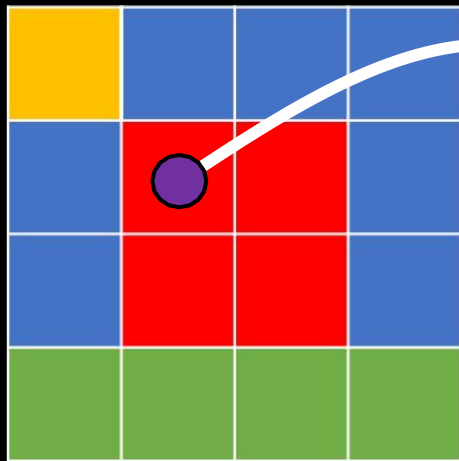


target

## Forward mapping: where do source pixels go?

0	0	0	0
0	1	1	1
0	1	1	1
0	0	0	0

disparity



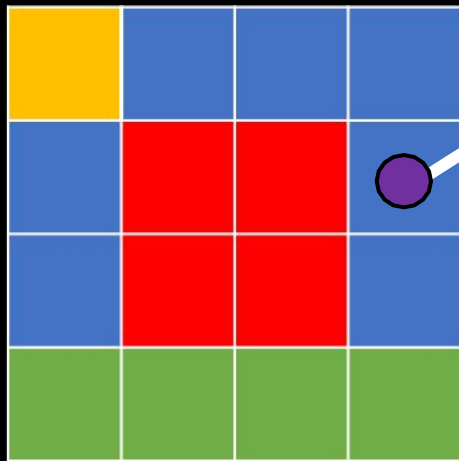
source

target

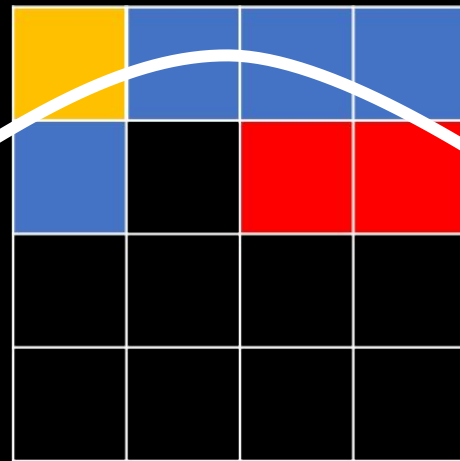
Forward mapping: where do source pixels go?

0	0	0	0
0	1	1	1
0	1	1	1
0	0	0	0

disparity



source

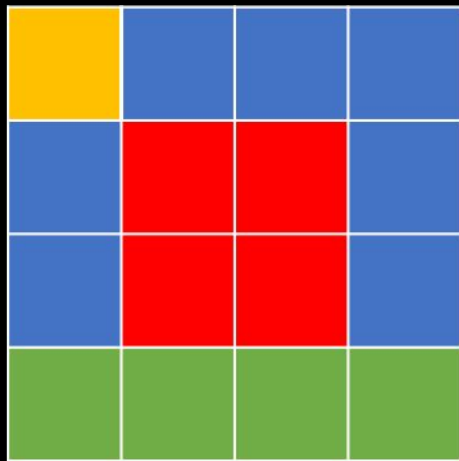


target

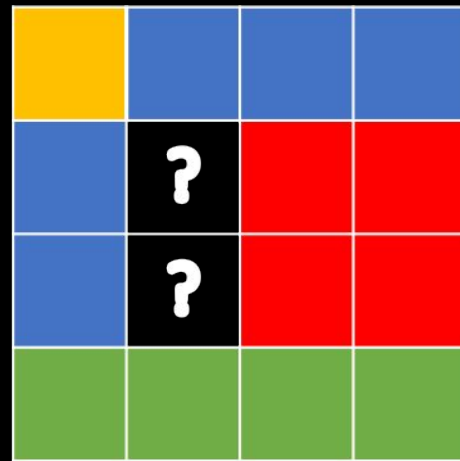
## Forward mapping: where do source pixels go?

0	0	0	0
0	1	1	1
0	1	1	1
0	0	0	0

disparity



source



target

Backward mapping: where do source pixels come from?

0	0	0	0
1	1	1	0
1	1	1	0
0	0	0	0

disparity


target

yellow	blue	blue	blue
blue	red	red	blue
blue	red	red	blue
green	green	green	green


source



















Backward mapping: where do source pixels come from?

0	0	0	0
1	1	1	0
1	1	1	0
0	0	0	0

disparity

target

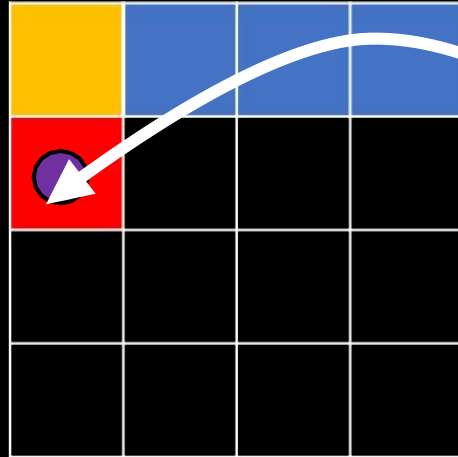
			
			
			
			

source

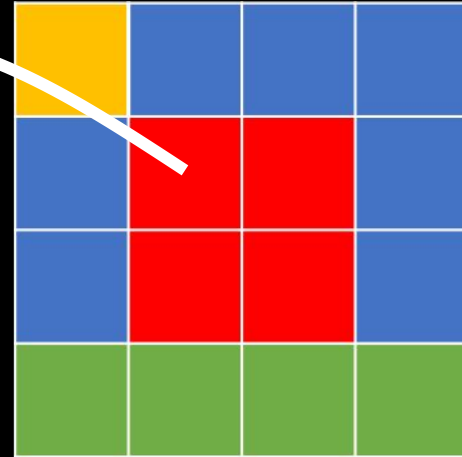
Backward mapping: where do source pixels come from?

0	0	0	0
1	1	1	0
1	1	1	0
0	0	0	0

disparity



target

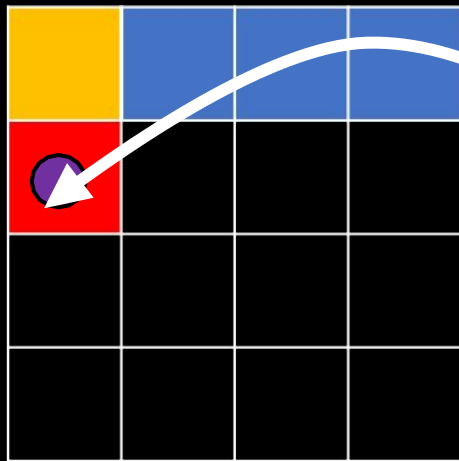


source

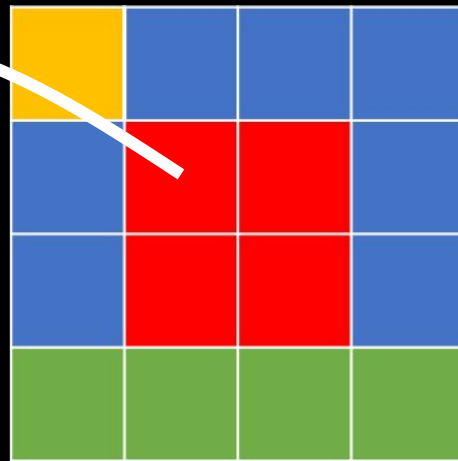
Backward mapping: where do source pixels come from?

0	0	0	0
1	1	1	0
1	1	1	0
0	0	0	0

disparity



target

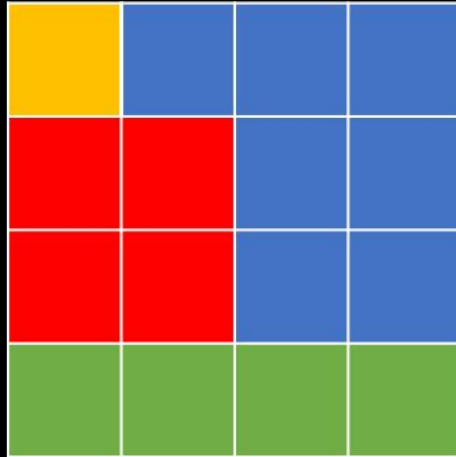


source

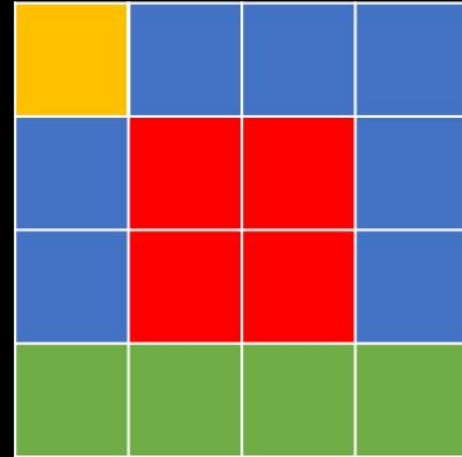
## Backward mapping: where do source pixels come from?

0	0	0	0
1	1	1	0
1	1	1	0
0	0	0	0

disparity



target



source

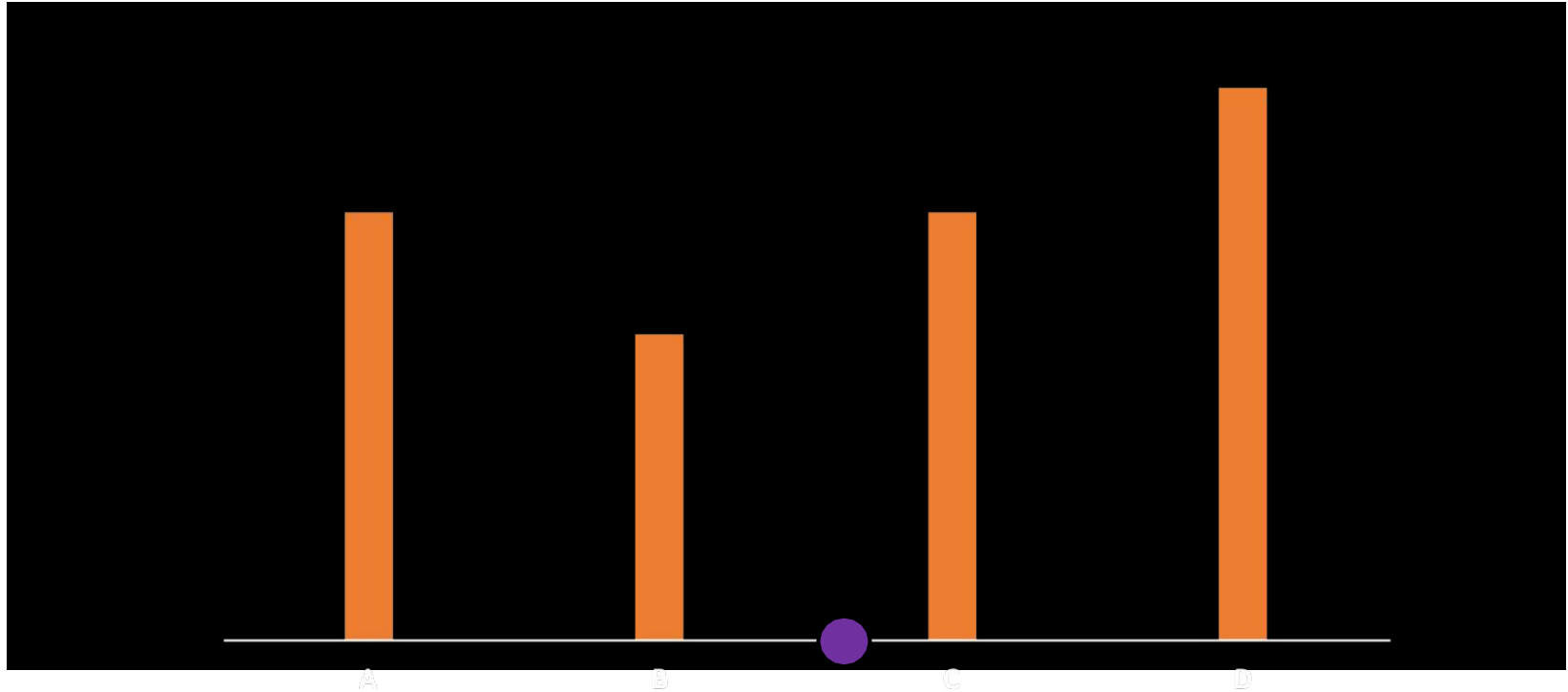
## Image sampling

<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
<b>1</b>	<b>2</b>	<b>0</b>	<b>0</b>
<b>3</b>	<b>1</b>	<b>0</b>	<b>0</b>
<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

## Image sampling

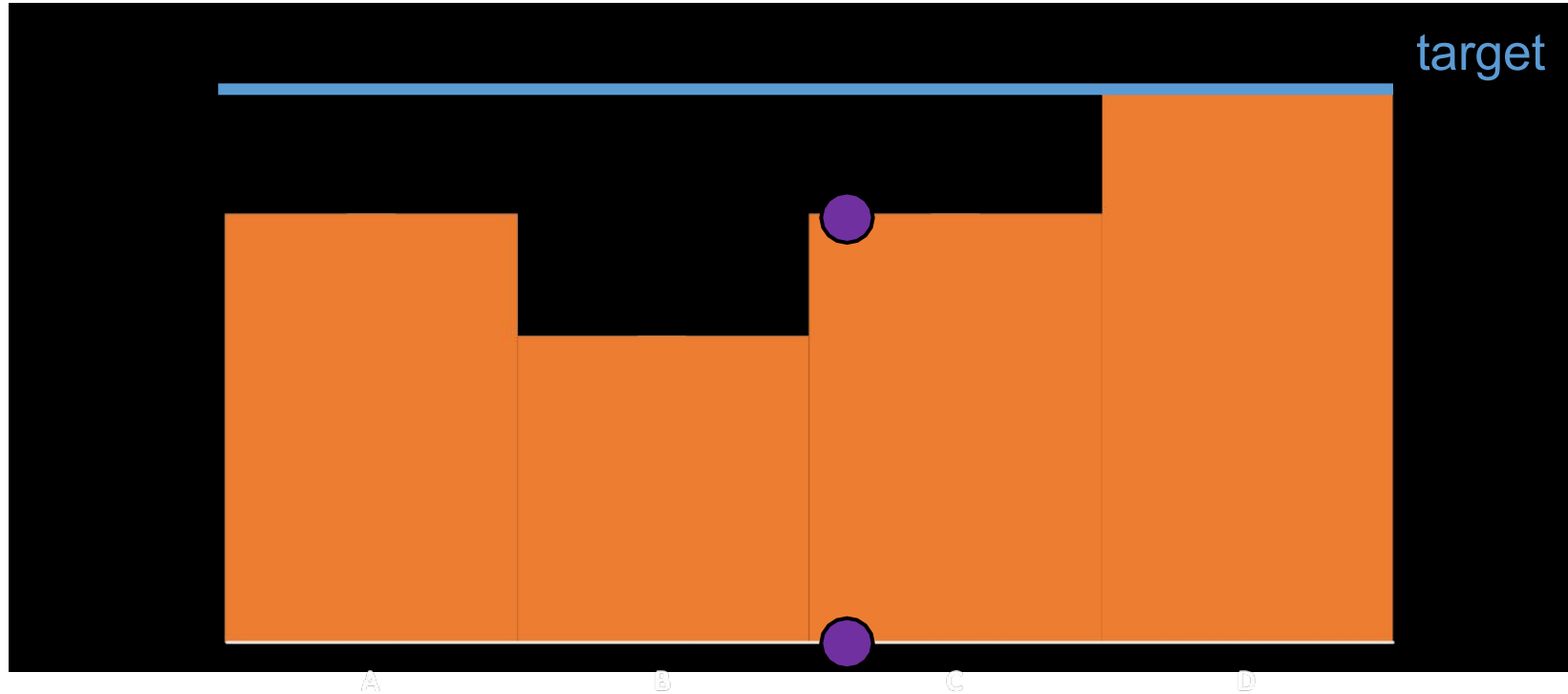
<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
<b>1.1</b>	<b>2.1</b>	<b>0</b>	<b>0</b>
<b>2.9</b>	<b>1.4</b>	<b>0</b>	<b>0</b>
<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

## Nearest neighbor sampling

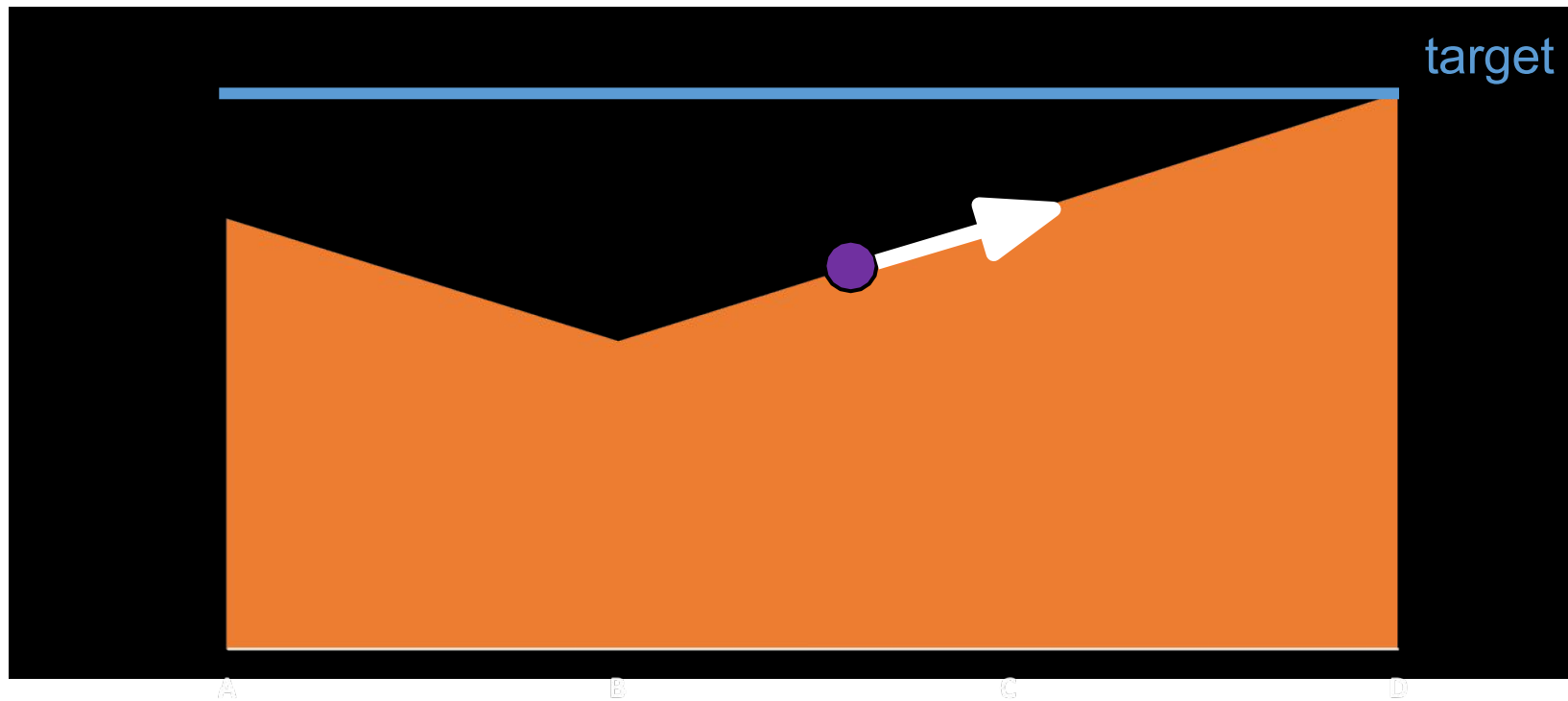




## Nearest neighbor sampling

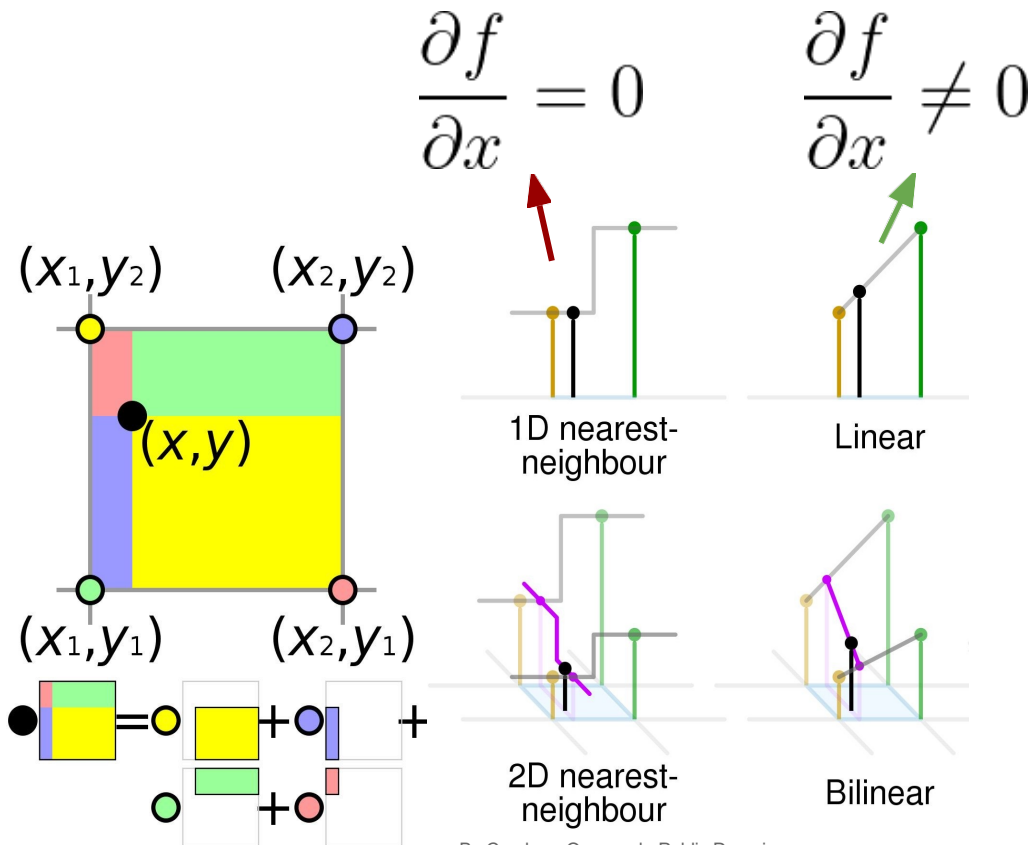


## Linear sampling



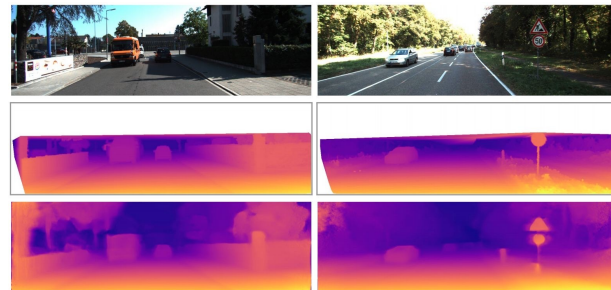
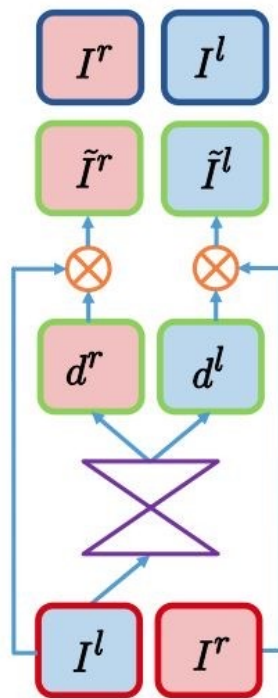
## Let's use Bilinear sampling!

- Spatial Transformer Networks [3] introduced bilinear sampling in a network at NeurIPS 2015
- Differentiable!
- In Pytorch
  - `torch.nn.functional.grid_sample`
- In TensorFlow
  - `tfa.image.dense_image_warp`



## Monodepth CVPR2017 [4]

- Use bilinear sampling to warp **R** to generate **L**
- Predicts both **L** and **R** disparities and enforces consistency between them
- **UNet** architecture with multiscale predictions
- Beat supervised methods at the time!



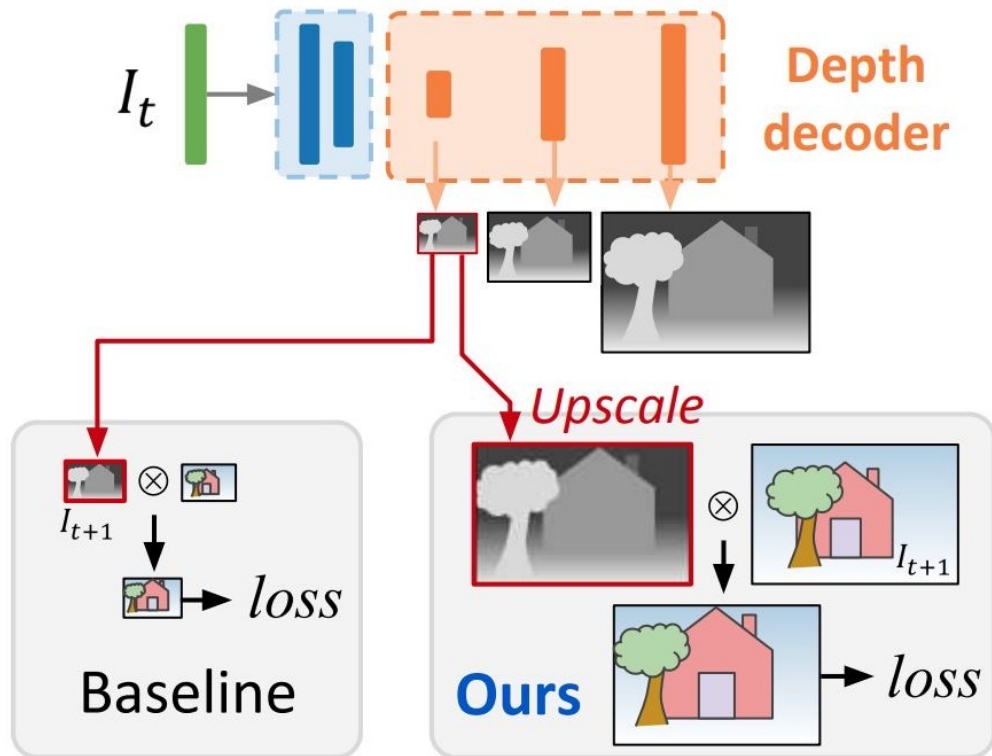
## Reconstruction loss

$$\frac{1}{N} \sum_{i,j} \alpha \frac{1 - \text{SSIM}(I_{ij}^l, \tilde{I}_{ij}^l)}{2} + (1 - \alpha) \|I_{ij}^l - \tilde{I}_{ij}^l\|.$$

- L1/L2 alone struggles to train
- L1 + dSSIM is the standard
  - SSIM favors texture reconstruction
- Feature loss
  - Warp pretrained features instead of RGB colors [5]

## Multiscale

- Coarse-to-fine depth prediction
- Reconstruction loss at each stage
- or Upsample depth *then* compute reconstruction loss at high res [6]



## Occlusions





## Occlusions



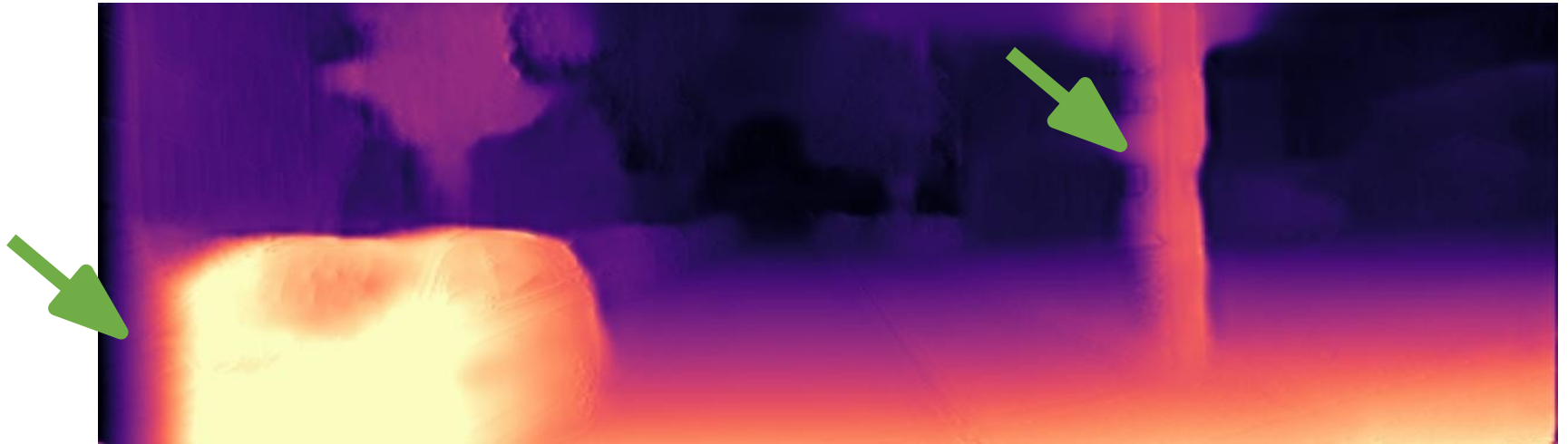
## Occlusions



## Occlusions



## Occlusions



## How to deal with occlusions

- Ignore them!
- Post processing [4]
- Predict an occlusion mask [7][8]
- Use virtual trinocular constraints [9]
- Feed Left or Right images randomly [6]



# Conclusion

We framed the depth prediction problem as an **image reconstruction** one.

Differentiable parametric image generation is easily achieved  
via **bilinear sampling**.

Good results are achieved using **multiscale, robust photometric** losses, and  
**UNet**-like architectures.



## References:

- 1: Xie, Junyuan, et al. "Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks." ECCV 2016.
- 2: Garg, Ravi, et al. "Unsupervised cnn for single view depth estimation: Geometry to the rescue." ECCV 2016.
- [3]: Jaderberg, Max, et al. "Spatial transformer networks." NeurIPS 2015.
- 4 : Godard, Clément, et al. "Unsupervised monocular depth estimation with left-right consistency." CVPR 2017.
- 5: Zhan, Huangying, et al. "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction." CVPR 2018
- 6: Godard, Clément, et al. "Digging into self-supervised monocular depth estimation." ICCV 2019.
- [7]: Zhou, Tinghui, et al. "Unsupervised learning of depth and ego-motion from video." CVPR 2017.
- [8]: Schellevis, Maarten. "Improving Self-Supervised Single View Depth Estimation by Masking Occlusion." arXiv 2019.
- [9]: Poggi, Matteo, et al. "Learning monocular depth estimation with unsupervised trinocular assumptions." 3DV 2018.



# Quiz

What is the relation between disparity and depth?