Deep Scene Understanding from Images

Matteo Poggi, Fabio Tosi, Pierluigi Zama Ramirez
Computer Vision Lab (CVLab), University of Bologna

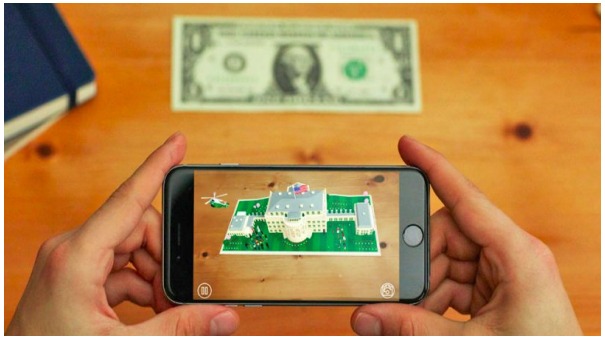
Monocular Depth Estimation

“Given a single RGB image as input, predict a dense depth map for each pixel ”

Monocular Depth Estimation - Motivation

- Using two or more cameras to triangulate the depth of the scene, although it increases the accuracy, it also adds more complexity: the cameras have to be **constantly recalibrated** due to the movement of the autonomous car or robot.

- Monocular techniques are an attractive solution for all those **low-cost** or **portable applications** where the use of multiple cameras would be **too costly** or **cumbersome**



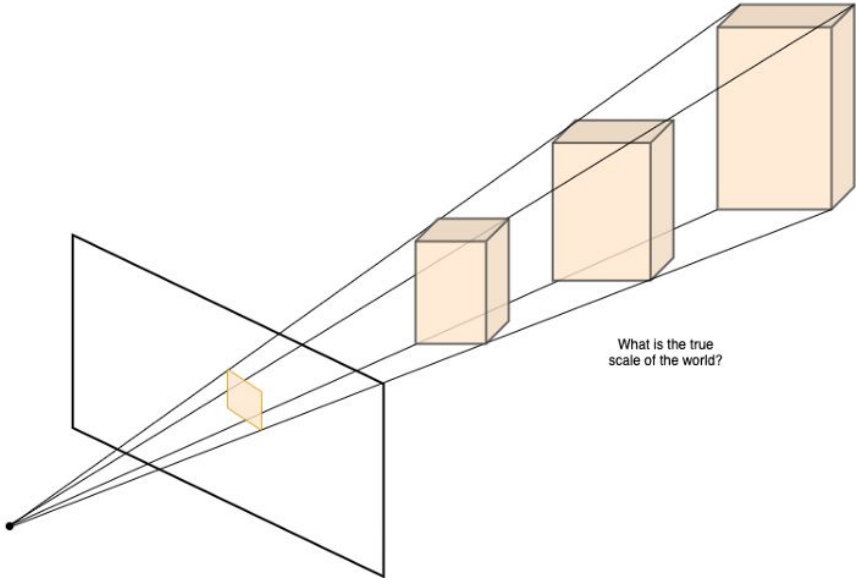
AR/VR

Surveillance System

Medical Application

Perspective Projection

- The **image formation** process deals with mapping a 3D space onto a 2D space
- Indeed, the mapping is **not a bijection**
- Estimating depth from a single image is an **<u>ill-posed</u>** problem



Perceiving 3D from 2D

- Humans excel at this task

Perceiving 3D from 2D

- Meaningful monocular cues:
  - ❑ **Linear Perspective**

"Linear perspective *is a depth cue that utilizes the fact that lines converge in the distance. That is,* ***parallel lines*** *will get "closer together" or* ***narrower*** *as they appear farther from the viewer."*

Perceiving 3D from 2D

- Meaningful monocular cues:
  - ❑ Linear Perspective
  - ❑ **Relative Size**

***"Closer** objects appears **larger** than objects further away. Therefore, if two objects are expected to be the same size, then the larger object will appear closer"*

Perceiving 3D from 2D

- Meaningful monocular cues:
  - ❑ Linear Perspective
  - ❑ Relative Size
  - ❑ **Interposition**

***"Interposition** involves objects that appear to be coming inbetween the viewer and another object. If an object is interfering with, or overlapping the sight of the second object, it is perceived closer "*

Perceiving 3D from 2D

- Meaningful monocular cues:
  - ❏ Linear Perspective
  - ❏ Relative Size
  - ❏ Interposition
  - ❏ **Texture Gradient**

*“When you're looking at an object that extends into the distance, such as a grassy field, the **texture** becomes less and less apparent the farther it goes into the distance”*

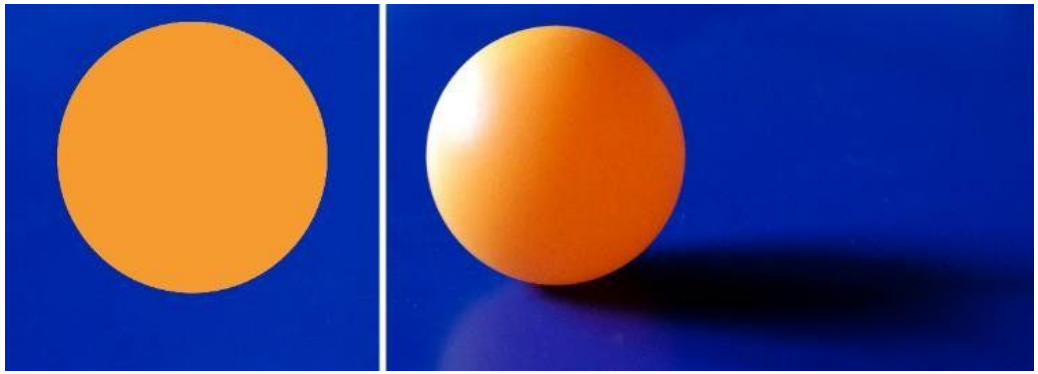
Perceiving 3D from 2D

- Meaningful monocular cues:
  - ❑ Linear Perspective
  - ❑ Relative Size
  - ❑ Interposition
  - ❑ Texture Gradient
  - ❑ **Height in plane**

*“In a picture, objects that are **further** from the viewer appear **higher** in the visual field. Likewise, **lower** objects suggest that they are **closer** to the viewer.”*

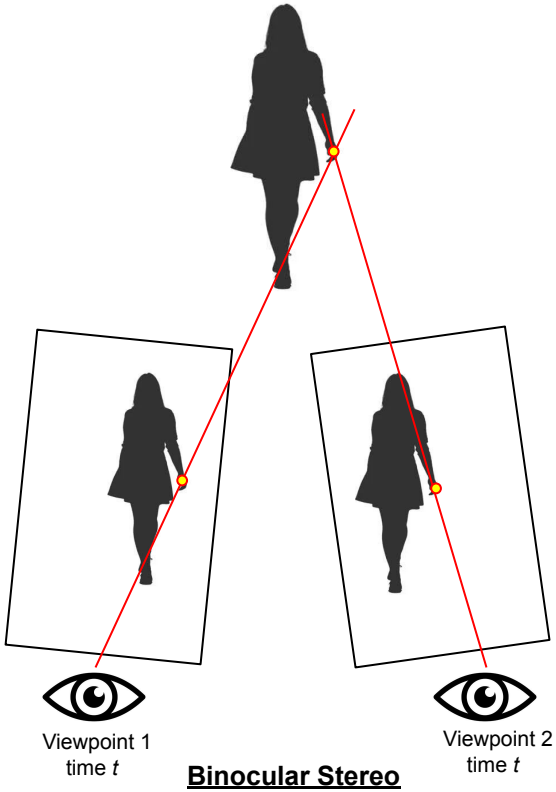
Perceiving 3D from 2D

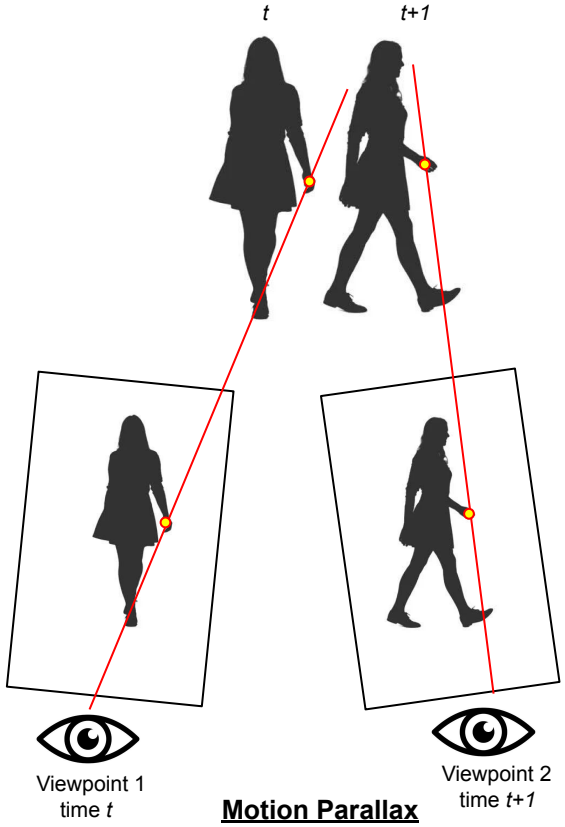
- Meaningful monocular cues:
  - ❑ Linear Perspective
  - ❑ Relative Size
  - ❑ Interposition
  - ❑ Texture Gradient
  - ❑ Height in plane
  - ❑ **Light and Shadow**



*“Patterns of **light** and **dark** can create the illusion of a three dimensional figure. This concept can be useful in judging distance.”*

- Additional powerful depth cues arise when a scene is viewed from multiple vantage points



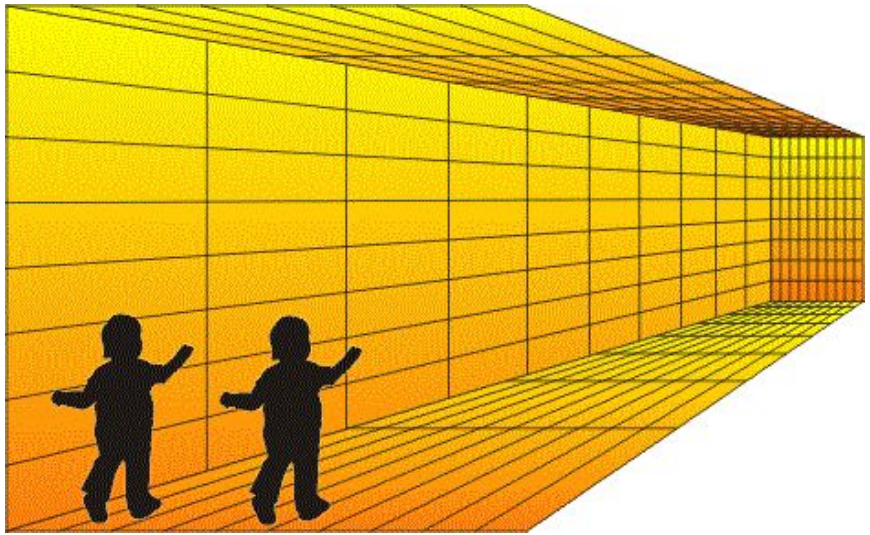


Optical illusions

- Monocular cues don’t always help:

Optical illusions

- Ponzo illusion (human mind judges an object's size based on its background)



https://www.eruptingmind.com/depth-perception-cues-other-forms-of-perception/

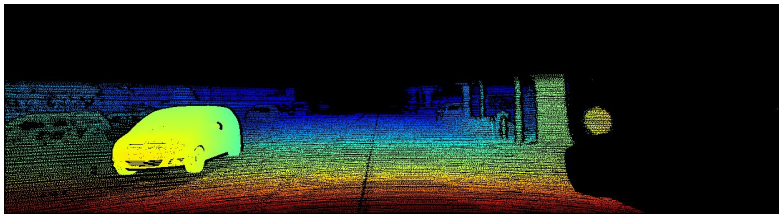
https://www.moillusions.com/these-3-cars-are-same-in-size/

Depth Estimation in Monocular Images

In Computer Vision, existing solutions to depth estimation from a single image usually rely on **deep learning** based approaches:

- **Supervised**
  - ground-truth depth data (RGB-D cameras, 3D laser scanners)
- **Semi-Supervised**
  - sparse ground-truth depth + image reconstruction
- **Self-Supervised**
  - image reconstruction (from monocular videos/stereo pairs/stereo sequences)
- **Proxy-Supervised**
  - depth labels extracted using external methods

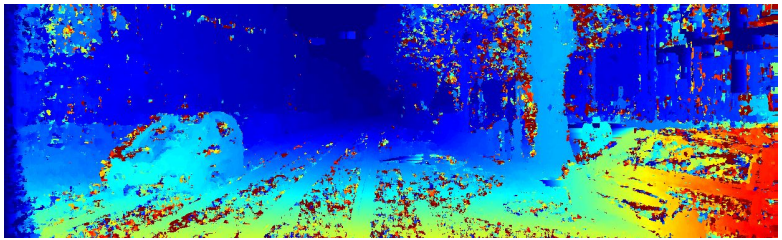
Depth Estimation in Monocular Images



Supervised (e.g. filtered LiDAR)

Semi-Supervised (raw LiDAR + images)

Self-Supervised (e.g. video or stereo)



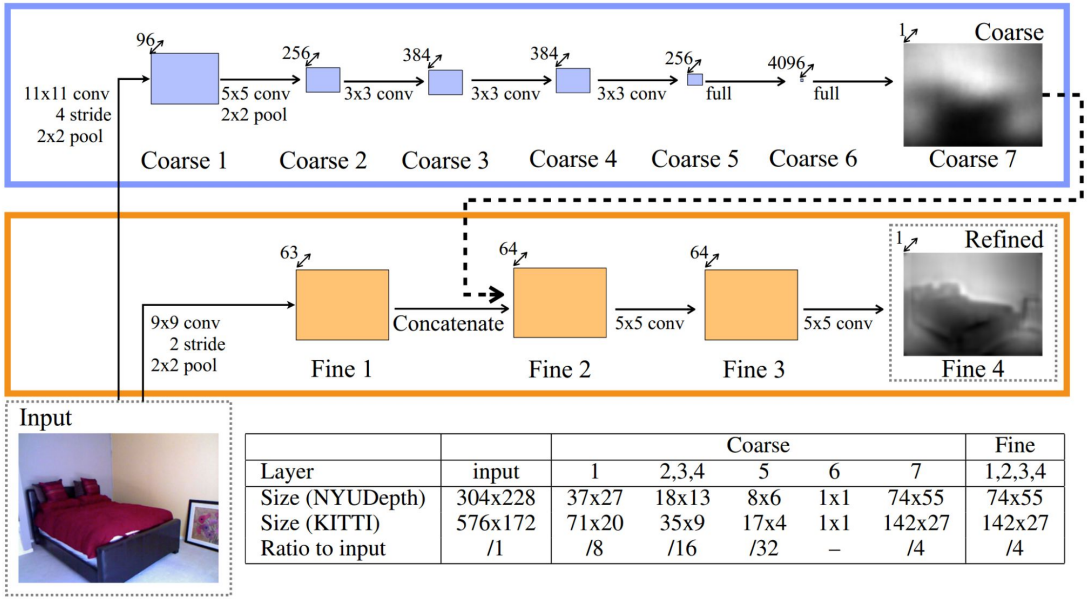
Proxy-Supervised (e.g. SGM)

Depth Map Prediction from a Single Image using a Multi-Scale Deep Network (Eigen, 2014)

- Neural network with two components:

1. one that first estimates the **global structure** of the scene
2. a second that refines it using **local information**

- Trained with ground-truth depth labels



		Coarse					Fine
Layer	input	1	2,3,4	5	6	7	1,2,3,4
Size (NYUDepth)	304x228	37x27	18x13	8x6	1x1	74x55	74x55
Size (KITTI)	576x172	71x20	35x9	17x4	1x1	142x27	142x27
Ratio to input	/1	/8	/16	/32	–	/4	/4

Unsupervised Learning of Depth and Ego-Motion from Video (Zhou, 2017)

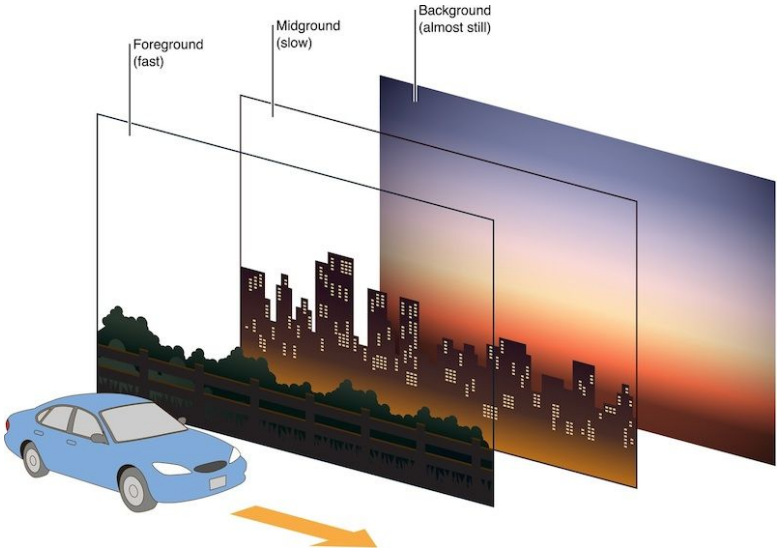
- Depth from **monocular images** and **ego motion**
- **Self-supervised** learning framework
- **End-to-end** learning approach

Video Sequence

Source Code: https://github.com/tinghuiz/SfMLearner

Motion Parallax

- When an observer **translates** relative to their visual environment, the **relative motion** of objects at different distances (motion parallax) provides a powerful cue to three-dimensional scene structure.



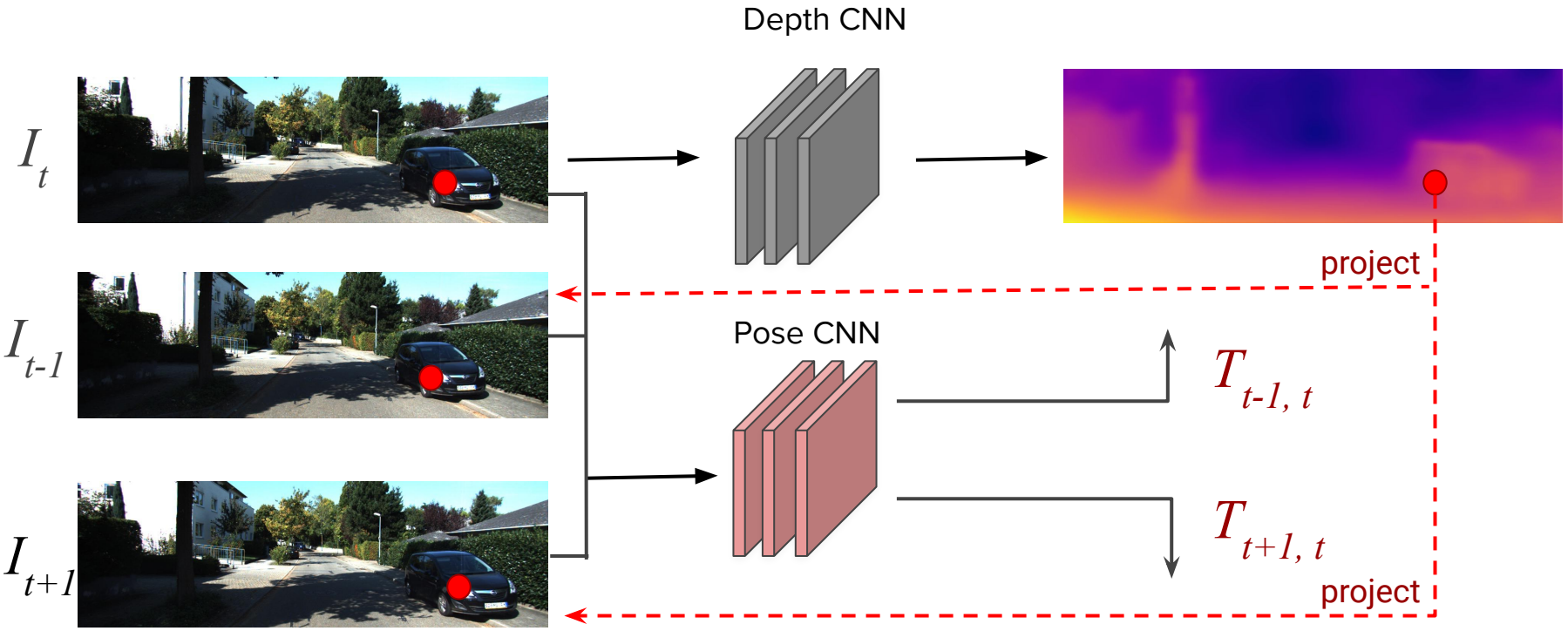
Motion Parallax

- When an observer **translates** relative to their visual environment, the **relative motion** of objects at different distances (motion parallax) provides a powerful cue to three-dimensional scene structure.

Unsupervised Learning of Depth and Ego-Motion from Video (Zhou, 2017)

- **Novel view synthesis** as key supervision signal
- Given one input view of a scene, synthesize a new image of the scene seen from a difference camera pose
- It is possible to synthesize a target view given:
  - **Depth** (for that image)
  - **Pose**
  - **Visibility** in a nearby view

View Synthesis as Supervision

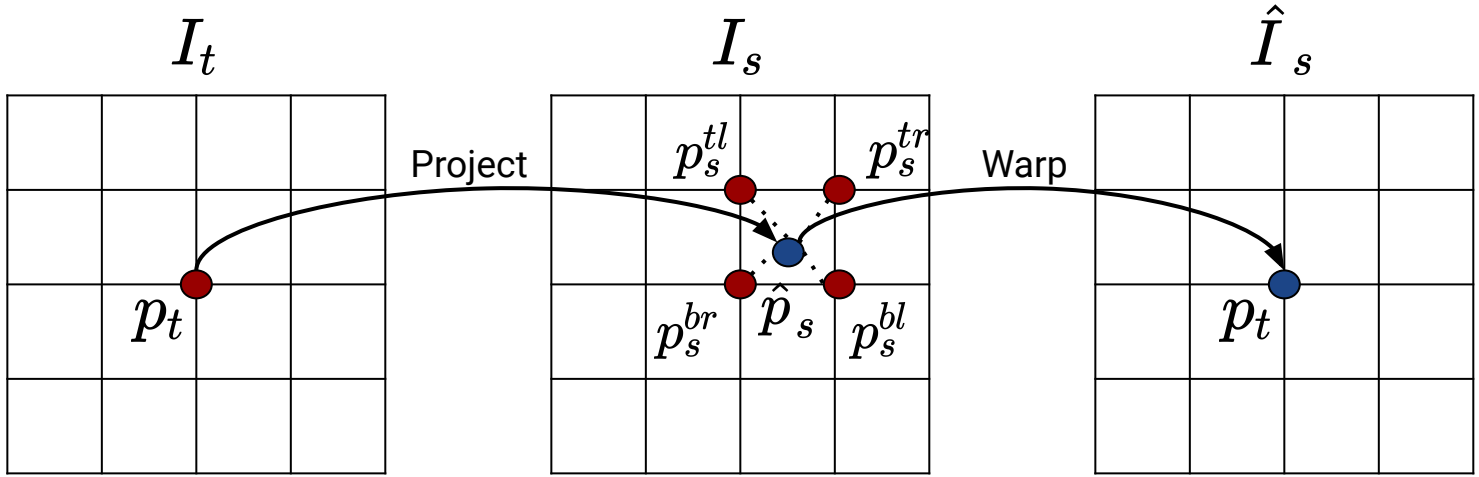


Assumptions

- The view synthesis formulation implicitly assumes:
  - Sufficient **illumination** in the environment
  - The scene is **static** without moving objects
  - Sufficient **motion parallax** in successive frames
  - Sufficient scene **overlap** between consecutive frames
  - There is **no occlusion** between the target view and the source views
  - The surface is **Lambertian**

- To improve the robustness:
  - *Explainability prediction network*

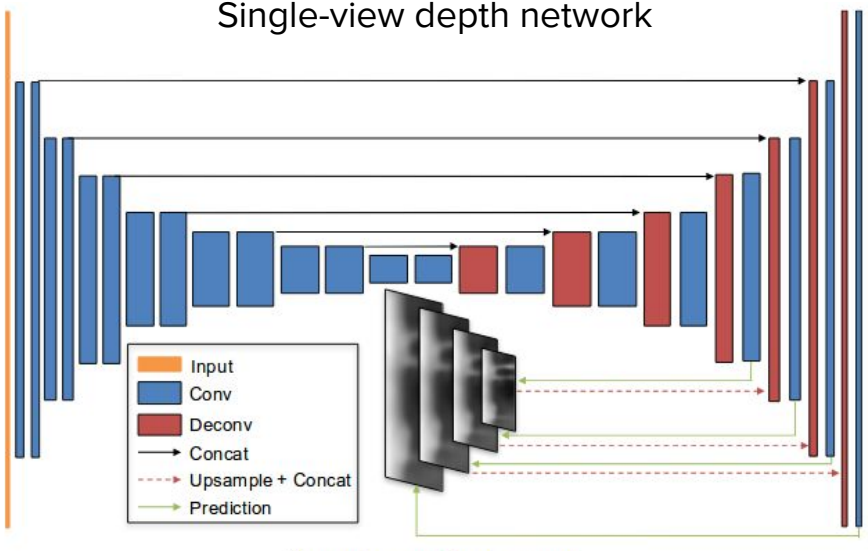
Differentiable depth image-based rendering (Jaderberg, 2015)



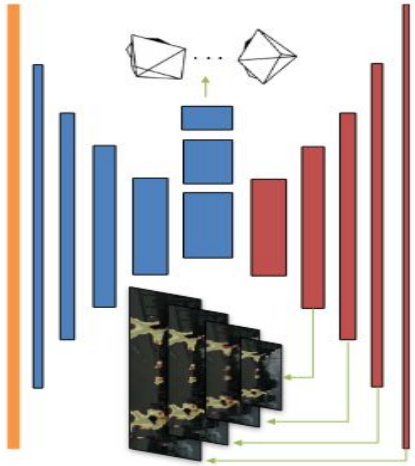
$I(\hat{p}_s)$ = bilinear interpolation of 4 neighbors

Network Architecture

Single-view depth network



Pose/explainability network



View synthesis loss

$$\mathcal{L}_{vs} = \sum_{<I_1,..,I_N>\in\mathcal{S}} \sum_{p} \hat{E}_s(p)|I_t(p) - \hat{I}_s(p)|$$

Total loss

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

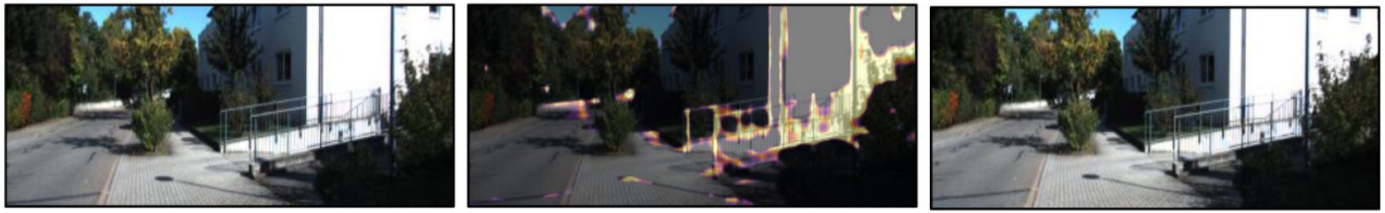
Explainability prediction →

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

- Dynamic objects

- Visibility/Occlusion

- Thin Structures



Target view | Explainability mask | Source view

Out-Of-View Pixels

- Out-of-view pixels due to **egomotion** at image boundaries

Time *t*

Out-Of-View Pixels

- Out-of-view pixels due to **egomotion** at image boundaries

Time *t+1*

Out-Of-View Pixels

- Out-of-view pixels due to **egomotion** at image boundaries

Time *t+1*

Out-Of-View Pixels

- The effect of out-of-view pixels can be reduced by **masking** such pixels in the reprojection loss. However, it **does not** handle **occluded regions** in the image

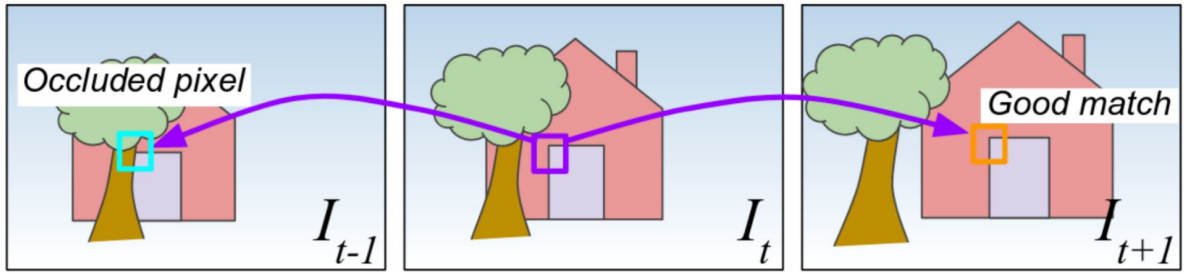
Time *t*

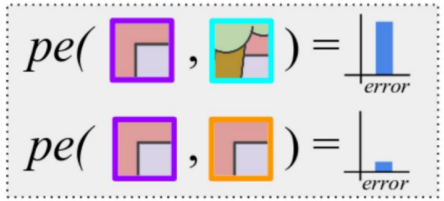
Warped Image

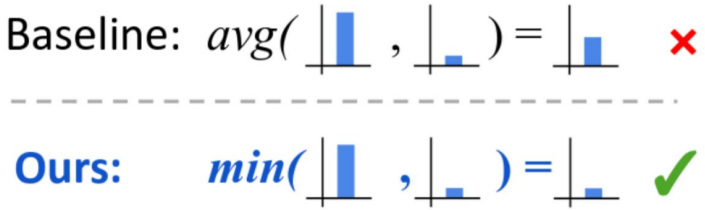
Mask

Handling Occlusions

- When the camera moves points in the scene that are visible in one frame may become **occluded** in another, and viceversa







Static and *"Car-following"* scenarios

- **Moving objects** and/or a **stationary camera** are an issue when training a single-image depth network on monocular video sequences

- If the moving object has the **same speed** and **direction** as the camera, then the reprojection error is low, i.e. a depth of infinity for that object.

Stationary camera (no parallax)

Static and *"Car-following"* scenarios

- **Moving objects** and/or a **stationary camera** are an issue when training a single-image depth network on monocular video sequences

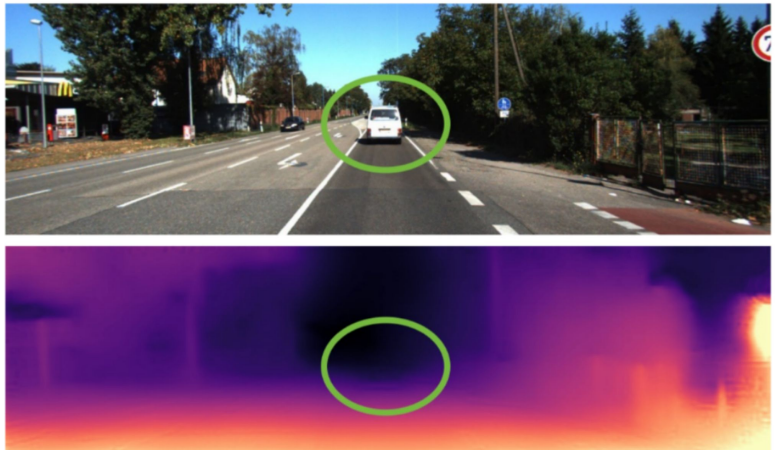
- If the moving object has the **same speed** and **direction** as the camera, then the reprojection error is low, i.e. a depth of infinity for that object.

Car following scenario

Static and *"Car-following"* scenarios

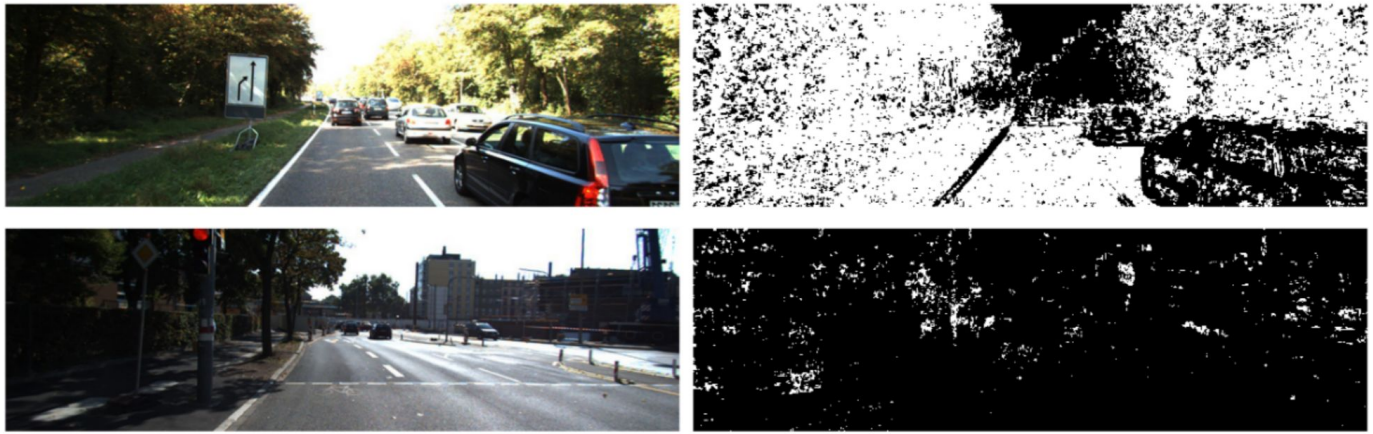
- **Moving objects** and/or a **stationary camera** are an issue when training a single-image depth network on monocular video sequences

- If the moving object has the **same speed** and **direction** as the camera, then the reprojection error is low, i.e. a depth of infinity for that object.



Static and *"Car-following"* scenarios

- **Auto-masking** as simple method that filters out pixels which **do not change** appearance from one frame to the next in the sequence

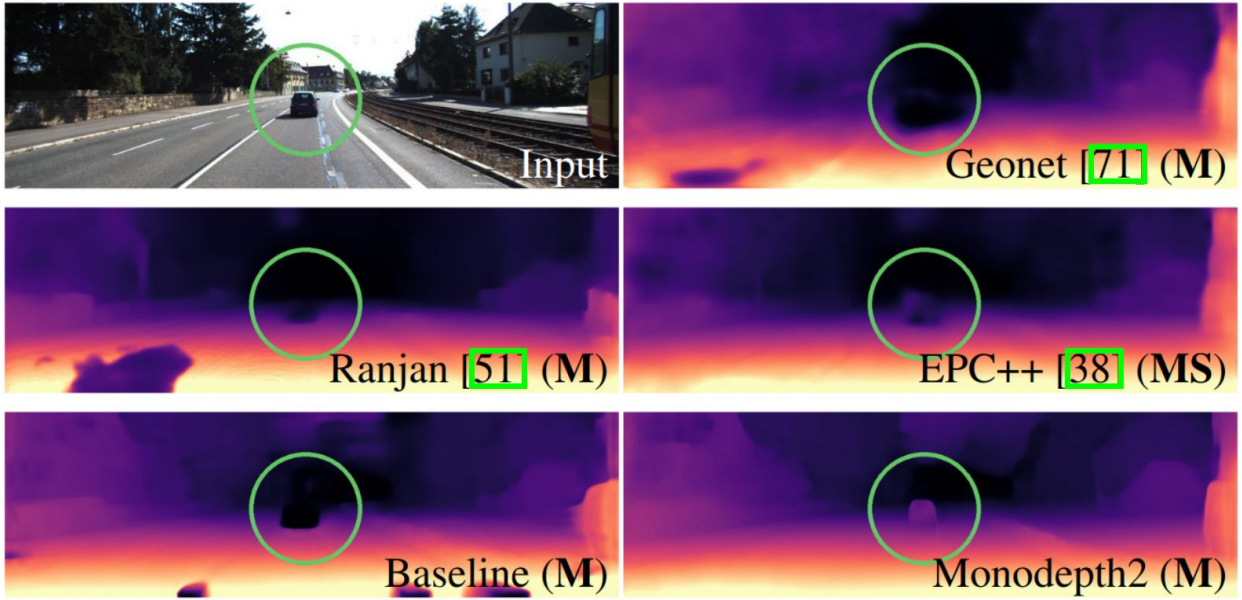


$$\mu = [\min_{ij} pe(I_t^{ij}, \hat{I}_s^{i,j}) < \min_{ij} pe(I_t^{ij}, I_s^{ij})$$

pe:per-pixel minimum

Static and *"Car-following"* scenarios

- **Auto-masking** as simple method that filters out pixels which **do not change** appearance from one frame to the next in the sequence



Unsupervised Monocular Depth Estimation with Left-Right Consistency (Godard, 2018)

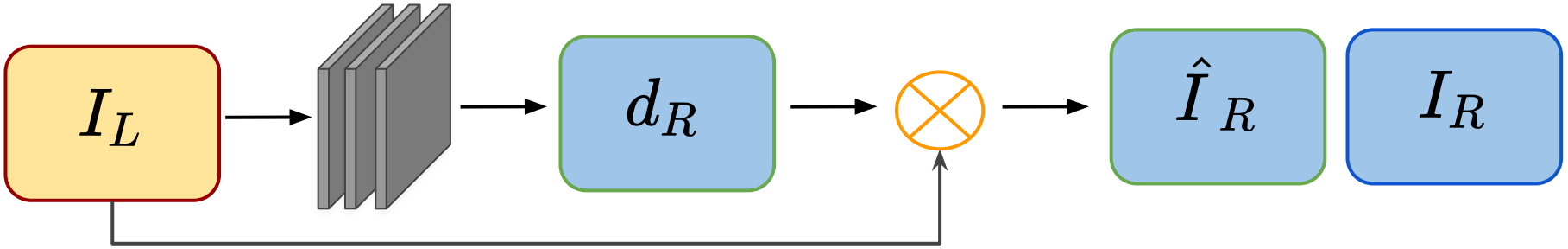
- Given a **calibrated stereo pair** at training time, the goal is to find a dense correspondence field (***disparity***) that, when applied to the left/right image, would enable to reconstruct the right/left image

- Given the predicted disparity, the baseline and the focal length, we can trivially recover the depth as:

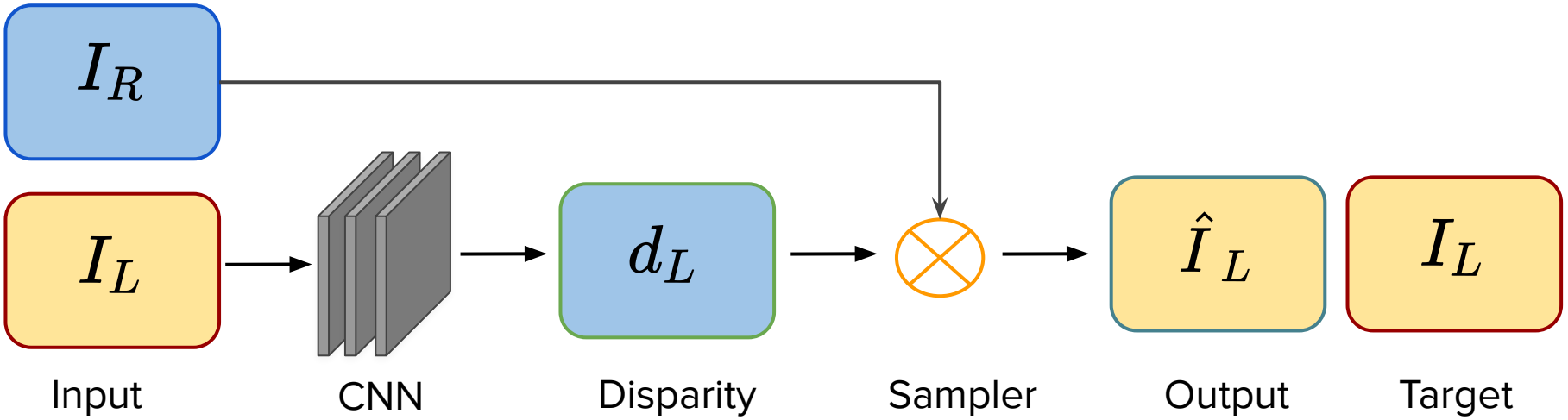
$$z = \frac{bf}{d}$$

Source Code: https://github.com/mrharicot/monodepth

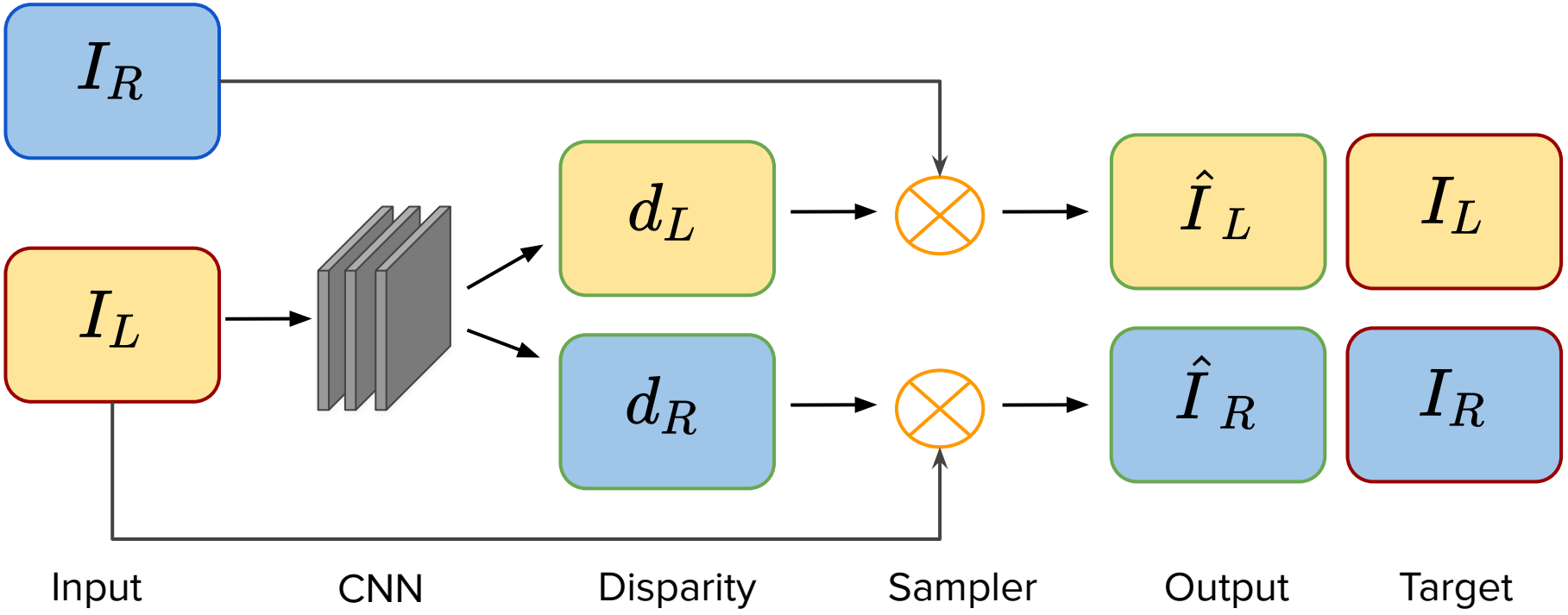
Naive approach



No Left-Right Consistency



Left-Right Consistency



Training Loss

Based on three comparison measurements: i) luminance ii) contrast and iii) structure

- Appearance Matching loss

$$\mathcal{L}_{ap}^{l} = \frac{1}{N}\sum_{ij}\alpha\frac{1-SSIM(I_{ij}^{l},\hat{I}_{ij}^{l})}{2} + (1-\alpha)||I_{ij}^{l} - \hat{I}_{ij}^{l}||$$

- Disparity Smoothness Loss

$$\mathcal{L}_{ds}^{l} = \frac{1}{N}\sum_{ij}|\partial_x d_{ij}^{l}|e^{-||\partial_x I_{ij}^{l}||} + |\partial_y d_{ij}^{l}|e^{-||\partial_y I_{ij}^{l}||}$$

- **Left-Right Disparity Consistency Loss**

$$\mathcal{L}_{lr}^{l} = \frac{1}{N}\sum_{ij}|d_{ij}^{l} - d_{ij+d_{ij}^{l}}^{r}|$$

- Total Loss

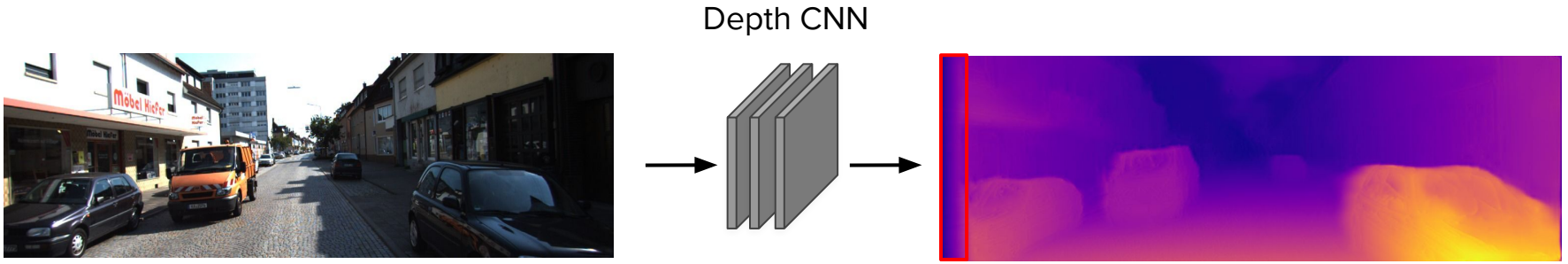
$$\mathcal{L}_{s}^{l} = \alpha_{ap}(\mathcal{L}_{ap}^{l} + \mathcal{L}_{ap}^{r}) + \alpha_{ds}(\mathcal{L}_{ds}^{l} + \mathcal{L}_{ds}^{r}) + \alpha_{lr}(\mathcal{L}_{lr}^{l} + \mathcal{L}_{lr}^{r})$$

Limitations for training on stereo images using image riprojection only

- Leveraging for training on stereo imagery yields state-of-the art performance
- In this way, the depth representation learned by the network is affected by artifacts in specific image regions (occlusions)

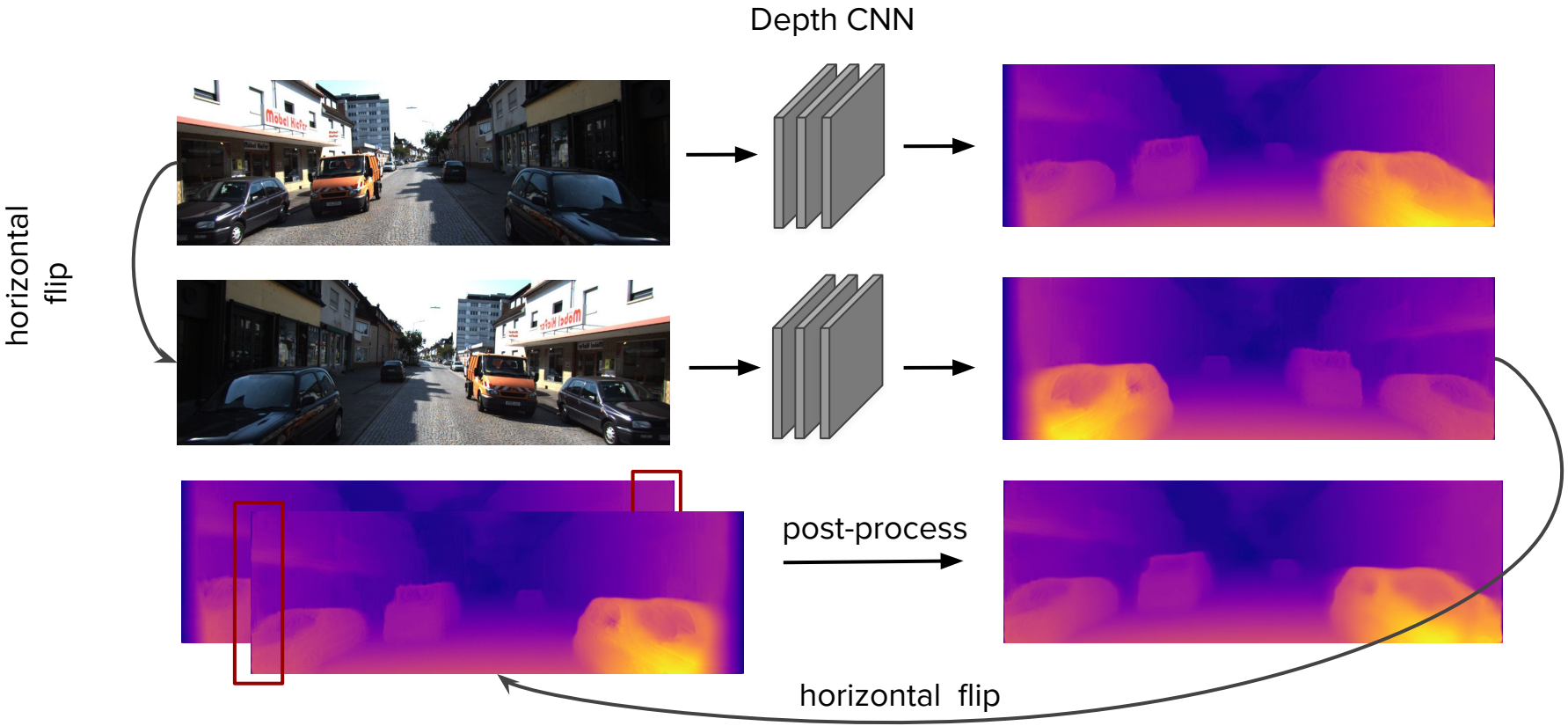
- Post-processing partially compensates for these artifacts, but it requires a double forward of the input image

Post-processing



- This way, the depth representation learned by the network is affected by **artifacts** in specific image regions **inherited from the stereo setup** (e.g., the left border using the left image as the reference and in occluded areas).
- A **post-processing** step partially compensates such artifacts

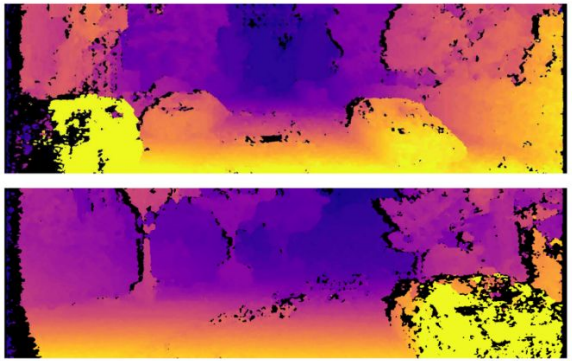
Post-processing



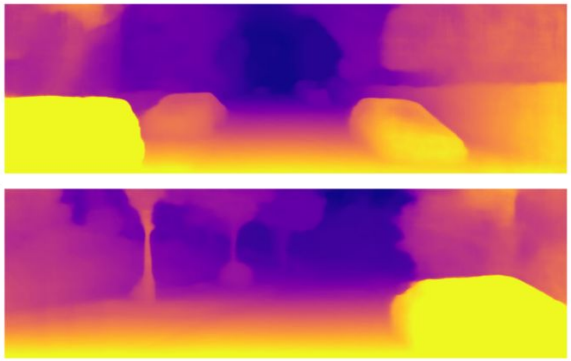
Learning monocular depth estimation infusing traditional stereo knowledge (Tosi, 2019)

- **Proxy** annotation through traditional stereo algorithms enables more accurate monocular depth estimation keeping a self-supervised approach

RGB



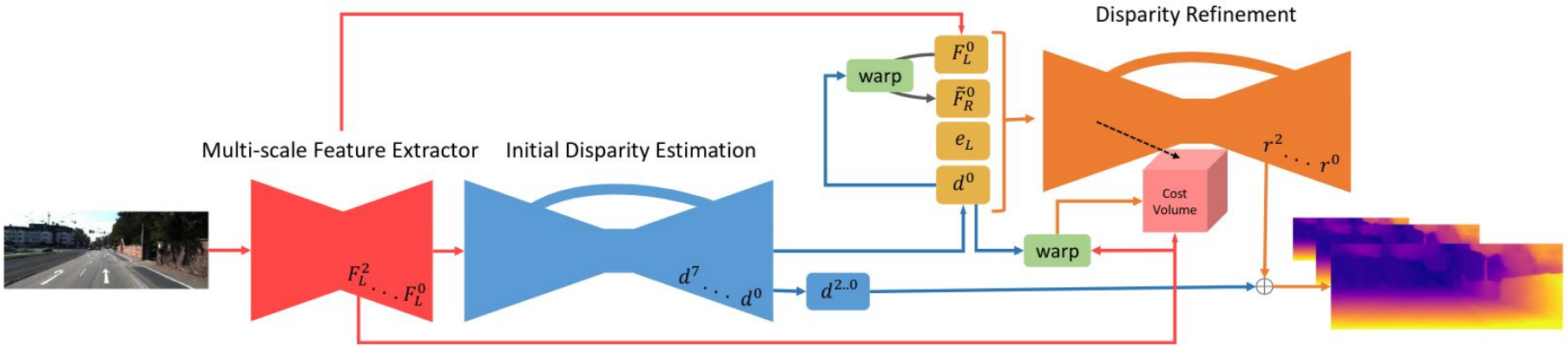
SGM



Monocular Network (monoResMatch)

Network Architecture

- A novel end-to-end architecture trained to estimate depth from a monocular image leveraging a virtual stereo setup



Self-Supervised Monocular Depth Hints (Watson, 2019)

- Existing self-supervised regression methods can struggle during training to find the **global optimum** when minimizing photometric reprojection loss
- **Depth hints** as depth suggestions to enhance an existing photometric loss function
- Depth hints can offer a more plausible reprojection
- Depth hints are only used, when needed, to **guide** the network out of the **local minima**

Training Image

Without Depth Hints

With Depth Hints

Training Image

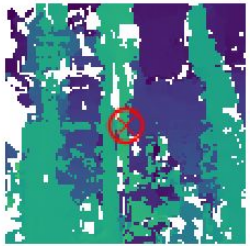
Without Depth Hints

With Depth Hints

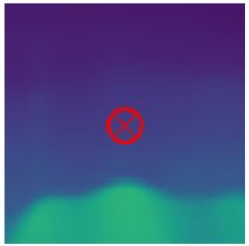
Image Patch

Other View

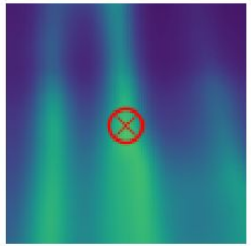
LiDAR



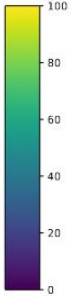
Fused SGM



Without Depth Hints



With Depth Hints



Colormap

Training Image

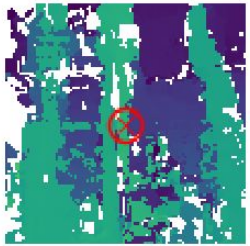
Without Depth Hints

With Depth Hints

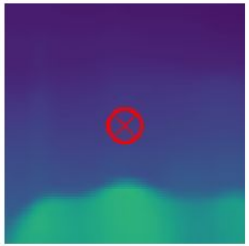
Image Patch

Other View

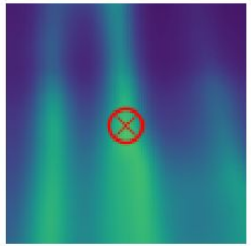
LiDAR



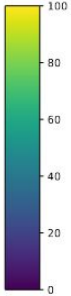
Fused SGM



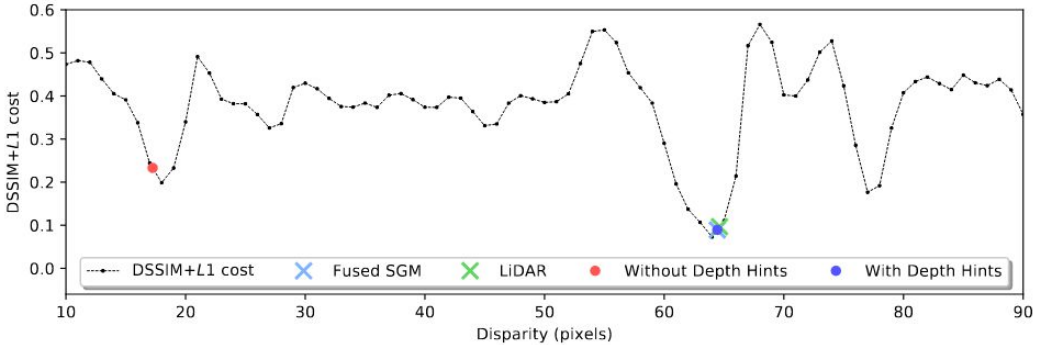
Without Depth Hints

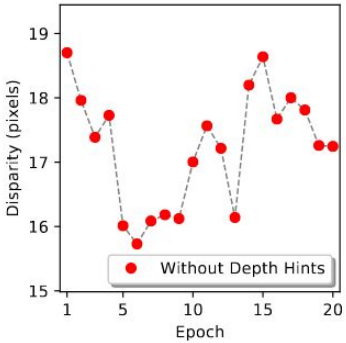


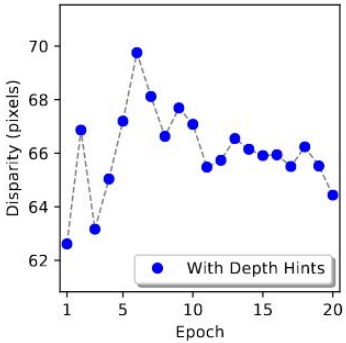
With Depth Hints



Colormap







- The goal is to optimize a given algorithm's existing loss, and to **consult** a pixel's depth hint only when the reprojection loss can be improved upon

- Total training loss

$$\mathcal{L} = \begin{cases} l_r(d_i) + l_s^{logL1}(d_i, h_i), & \text{if } l_r(h_i) < l_r(d_i). \\ l_r(d_i), & \text{otherwise} \end{cases}$$

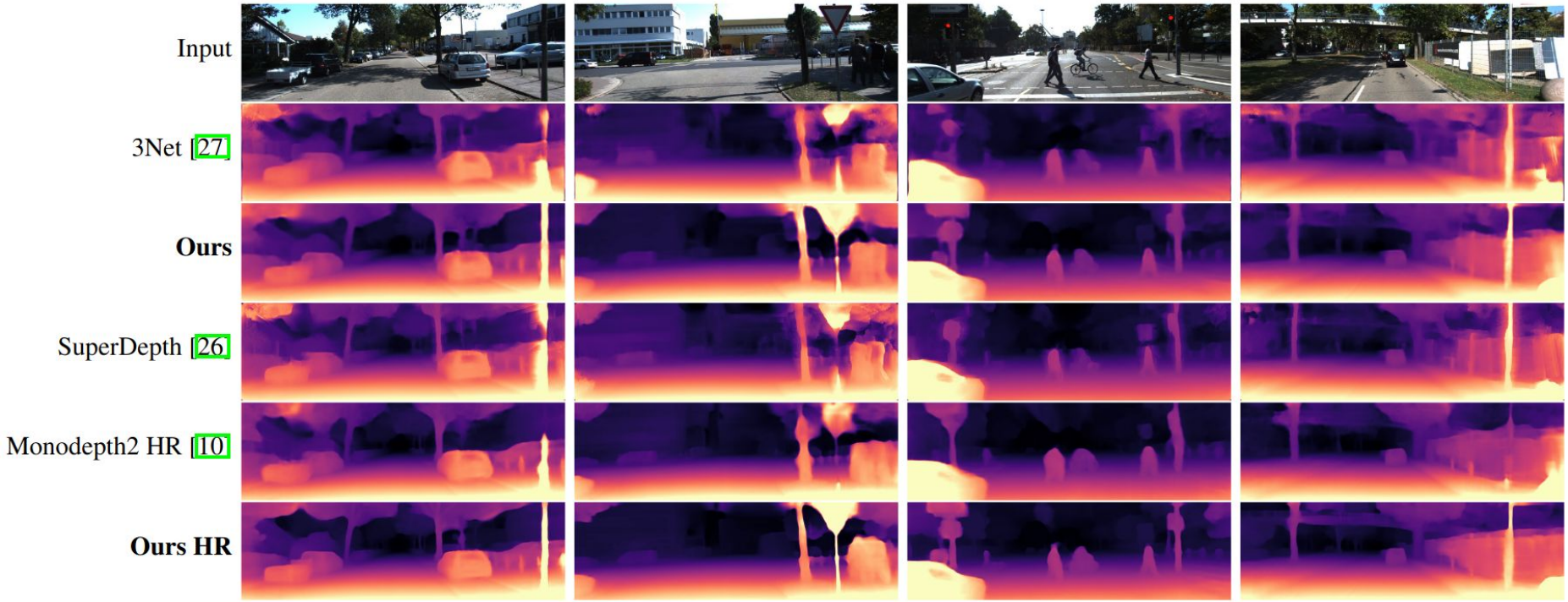
where

$$l_s^{logL1}(d_i, d_i') = log(1 + |d_i - d_i'|)$$

$$l_r(d_i) = \alpha \frac{1 - SSIM(I_i, \widetilde{I}_i)}{2} + (1 - \alpha)|I_i - \widetilde{I}_i|$$

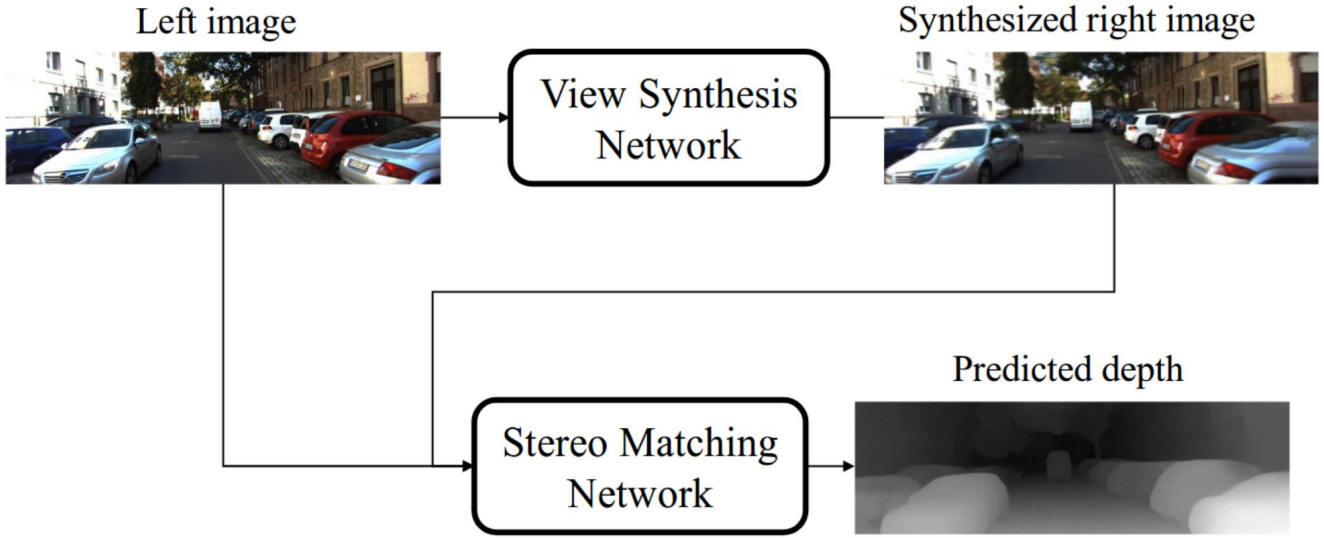
- Depth Hints extracted using a traditional stereo algorithm (e.g. SGM) on rectified **stereo pairs**

Qualitative Results

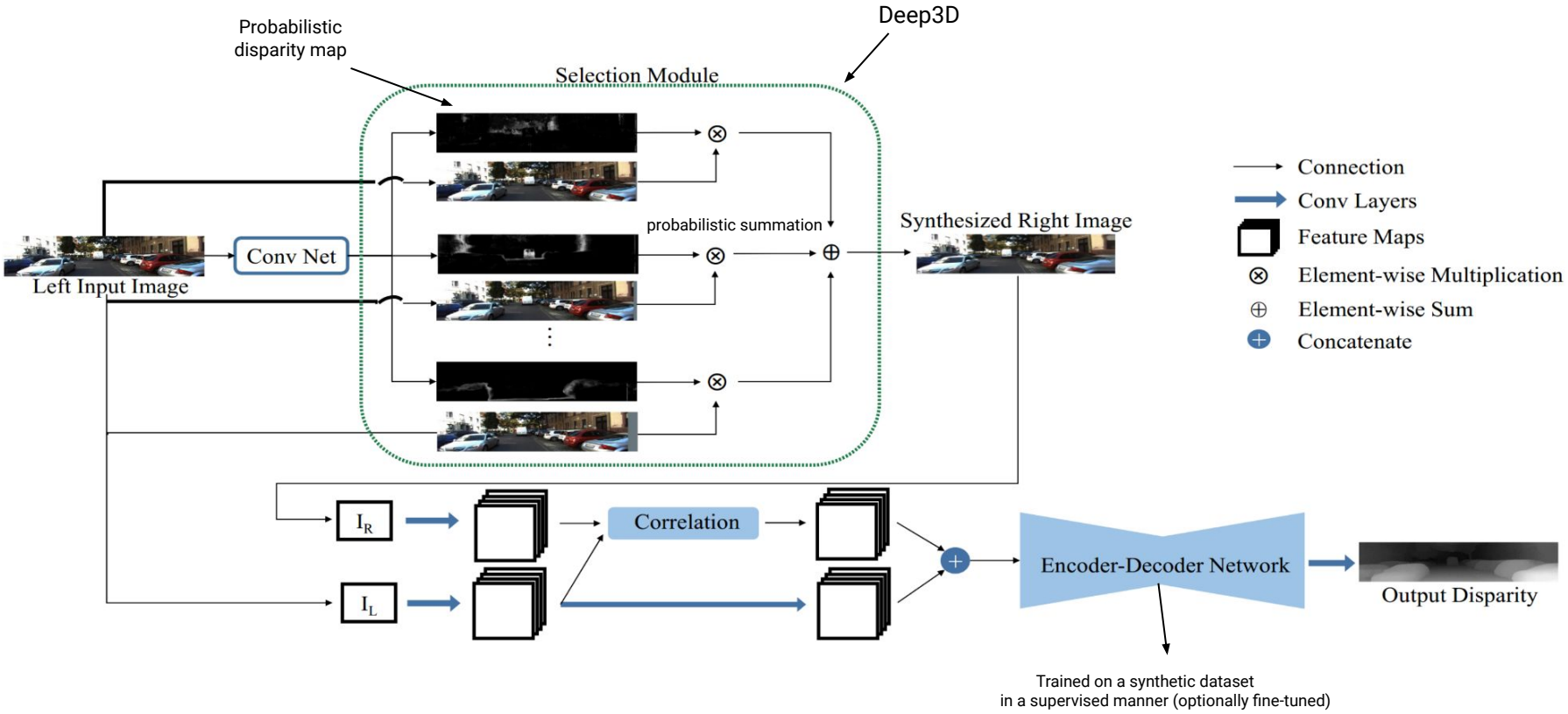


Single View Stereo Matching (Luo, 2018)

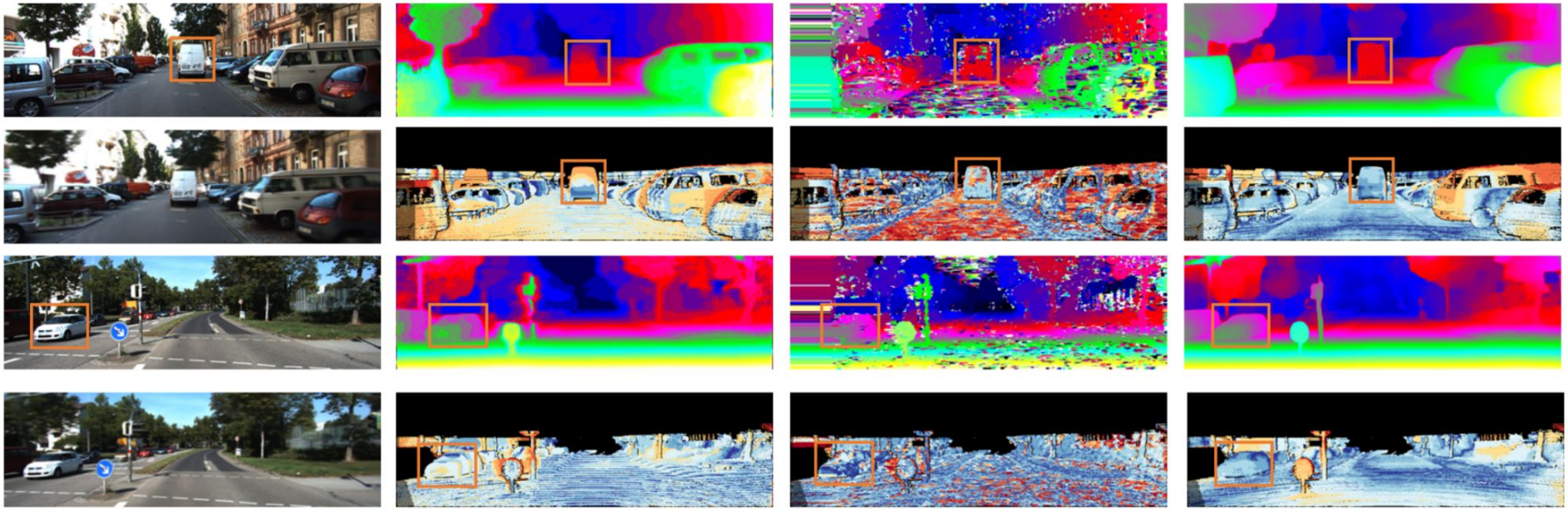
- Monocular depth estimation problem can be reformulated as two sub-problems, a **view synthesis** procedure followed by **stereo matching**



Framework


Probabilistic disparity map
Deep3D
Selection Module
probabilistic summation
Synthesized Right Image
Conv Net
Left Input Image
I_R
I_L
Correlation
Encoder-Decoder Network
Output Disparity
Trained on a synthetic dataset
in a supervised manner (optionally fine-tuned)
Connection
Conv Layers
Feature Maps
Element-wise Multiplication
Element-wise Sum
Concatenate

Qualitatives on the KITTI Benchmark



Input Image | Godard et al. | OCV-BM | Single-View Stereo (SVS)

Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer (MiDaS) [7]

https://github.com/intel-isl/MiDaS

How to obtain more training data?

- Deep learning methods have recently driven significant progress in the monocular depth estimation task.
- **RGB + Depth** as the best solution for training single-view deep learning based models but they are difficult to collect
- However, we need **training data** that captures the diversity of the visual world in order to obtain models that are effective across a **variety of scenarios**
- The key challenge is to acquire such data at sufficient scale

- Novel ways to train robust monocular depth estimation models that are expected to perform across diverse environments

- Experiments with **five diverse training datasets** (ReDWeb, MegaDepth, WSVD, DIML Indoor), including a new massive data source: **3D movies.** Each single dataset comes with its own characteristics and has its own biases and problems

- Tools that enable mixing multiple datasets during training, even if their annotations are incompatible

Dataset	Indoor	Outdoor	Dynamic	Video	Dense	Accuracy	Diversity	Annotation	Depth	# Images
DIML Indoor	✓			✓	✓	Medium	Medium	RGB-D	**Metric**	220K
MegaDepth		✓	(✓)		(✓)	Medium	Medium	SfM	No scale	130K
ReDWeb	✓	✓	✓		✓	Medium	**High**	Stereo	No scale & shift	3600
WSVD	✓	✓	✓	✓	✓	Medium	**High**	Stereo	No scale & shift	1.5M
3D Movies	✓	✓	✓	✓	✓	Medium	**High**	Stereo	No scale & shift	75K

Depth from 3D movies - Challenges

- The primary objective when producing stereoscopic film is providing a visually pleasing viewing experience while avoiding discomfort for the viewer: **disparity range is limited** and depends on both artistic and psychophysical considerations

- Focal lengths, baseline and convergence angle between the cameras of the stereo rig are **unknown** and vary between scenes

- In contrast to the standard stereo case, stereo pairs in movies usually contain both **positive** and **negative** disparities to allow objects to be perceived either in front or behind the screen

- Movies have **varying aspect ratios**, resulting in black bars on the top and bottom of the frame

Depth from 3D movies - Disparity Extraction

- To alleviate these problems, an **optical flow network** is applied to the stereo pairs
- The **horizontal component** of the flow as a proxy for disparity
- Optical flow **naturally handle** both positive and negative disparities
- **Left-right consistency check** and mark pixels with a disparity difference of more than 2 pixels as invalid.
- Other filtering procedures:
  - Frames are rejected if more than 10% of all pixels have a vertical disparity >2 pixels
  - Detect pixels that belong to sky regions using a pre-trained semantic segmentation model
  - Center crop to remove black bars

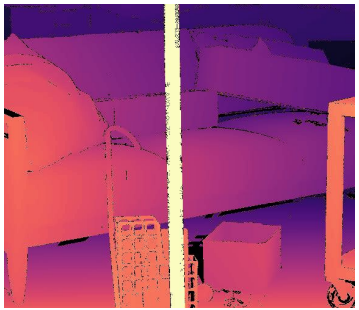
Depth from 3D movies

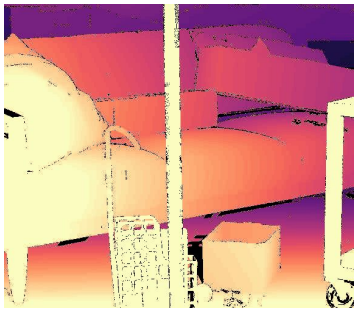
Training on Diverse Data - Challenges

1. Inherently different representations of depth: **Direct** vs **Inverse Depth**
2. **Scale ambiguity:** for some data (eg. depth from MVS), depth is only given up to an unknown scale
3. **Shift ambiguity**: some datasets provide disparity only up to an unknown scale and global disparity shift (e.g. 3D movies)


Reference Image


Direct depth


Inverse depth


Inverse depth + Shift ambiguity

Scale and shift invariant losses

d: prediction
d^*: groundtruth
$\hat{d}$: scaled and shifted prediction
$\hat{d}^*$: scaled and shifted groundtruth
ρ : loss function

- Prediction in disparity space

$$\mathcal{L}_{ssi}(\hat{d} - \hat{d}^*) = \frac{1}{2M} \sum_{i=1}^{M} \rho(\hat{d}_i - \hat{d}_i^*)$$

- Two strategies for alignment

1. **Least square criterion** + **mean squared error** (not robust for outliers)

estimators of scale and shift

$$(s,t) = \underset{(s,t)}{\operatorname{argmin}} \sum_{i=1}^{M} (sd_i + t - d_i^*)^2$$

s and *t* efficiently determined in closed form

2. **Robust estimators of scale and shift + absolute deviation**

$$\hat{d} = \frac{d - t(d)}{s(d)} \qquad \hat{d}^* = \frac{d^* - t(d^*)}{s(d^*)}$$

prediction and gt scaled to have zero translation and unit scale

$$s(d) = \frac{1}{M} \sum_{i=1}^{M} |d - t(d)| \qquad t(d) = median(d)$$

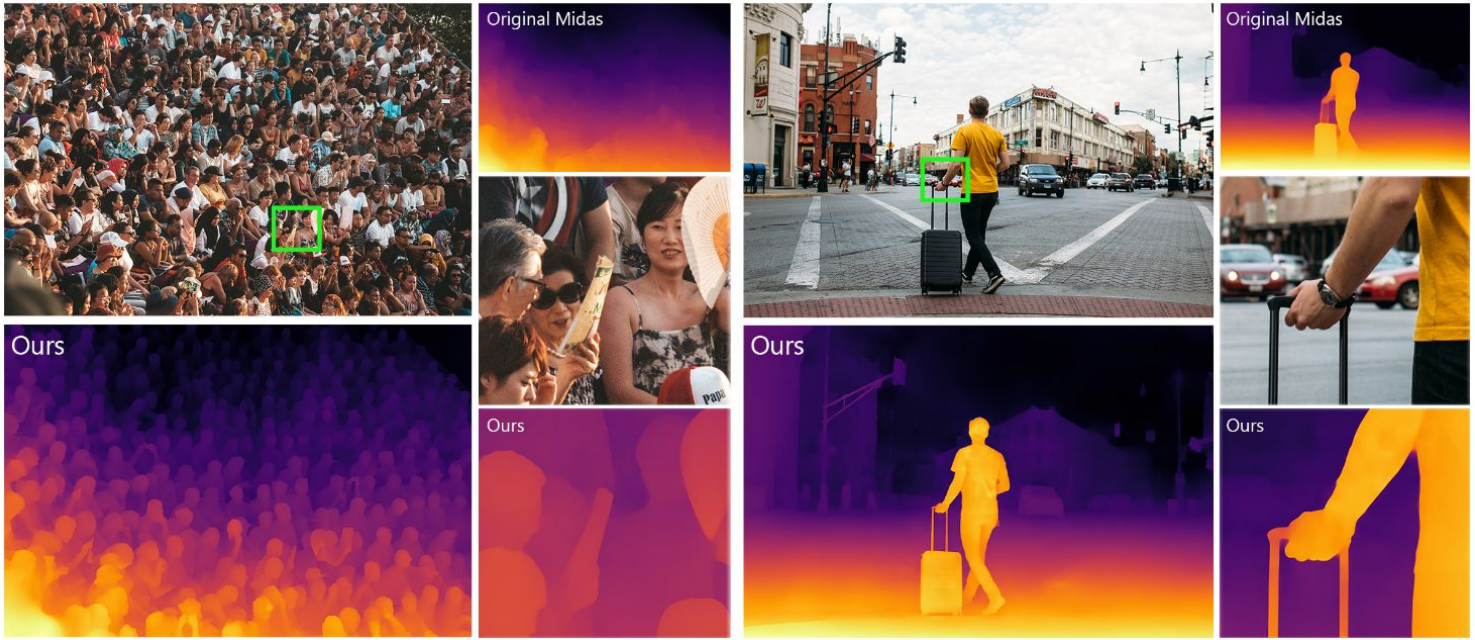
Qualitatives Examples

Qualitatives on Paintings and Drawings

Failure Cases

- Paintings, photos, and mirrors are often not recognized as such
- The model fails to recover the ground plane, likely because the input image was **rotated** by 90 degrees (data augmentation can be helpful)

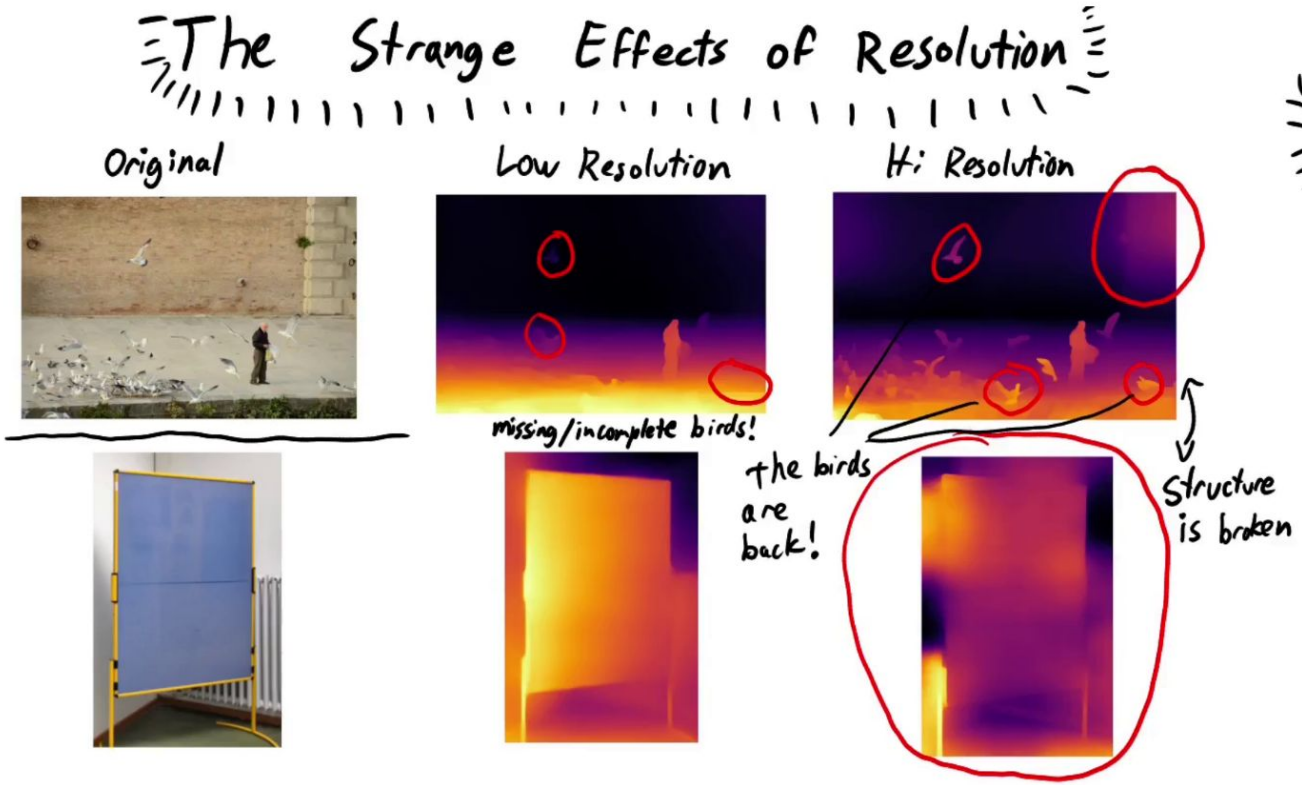
Boosting Monocular Depth Estimation Models to High-Resolution (Miangoleh, CVPR 2021)

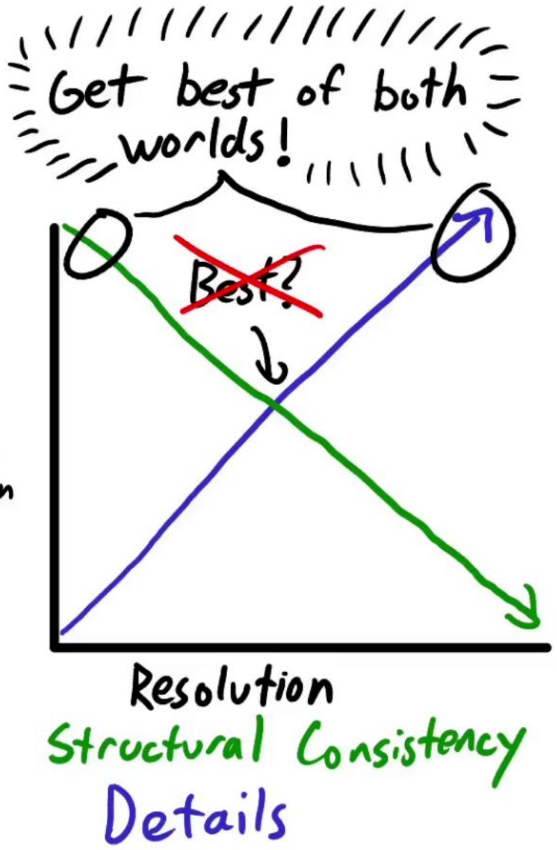


http://yaksoy.github.io/highresdepth/

Observations

- Depth maps extracted from standard monocular networks are well below one-megapixel resolution and often lack fine-grained details, which limits their practicality
- Practical constraints such as available GPU memory, lack of diverse high-resolution datasets, and the receptive field size of CNN's limit the potential of current methods
- The output characteristics of monocular depth estimation networks change with the resolution of the input image:
    - **Low Resolution**: the estimations have a consistent structure while lacking high-frequency details
    - **High Resolution**: the high-frequency details are captured much better while the structural consistency of the estimated depth gradually degrades

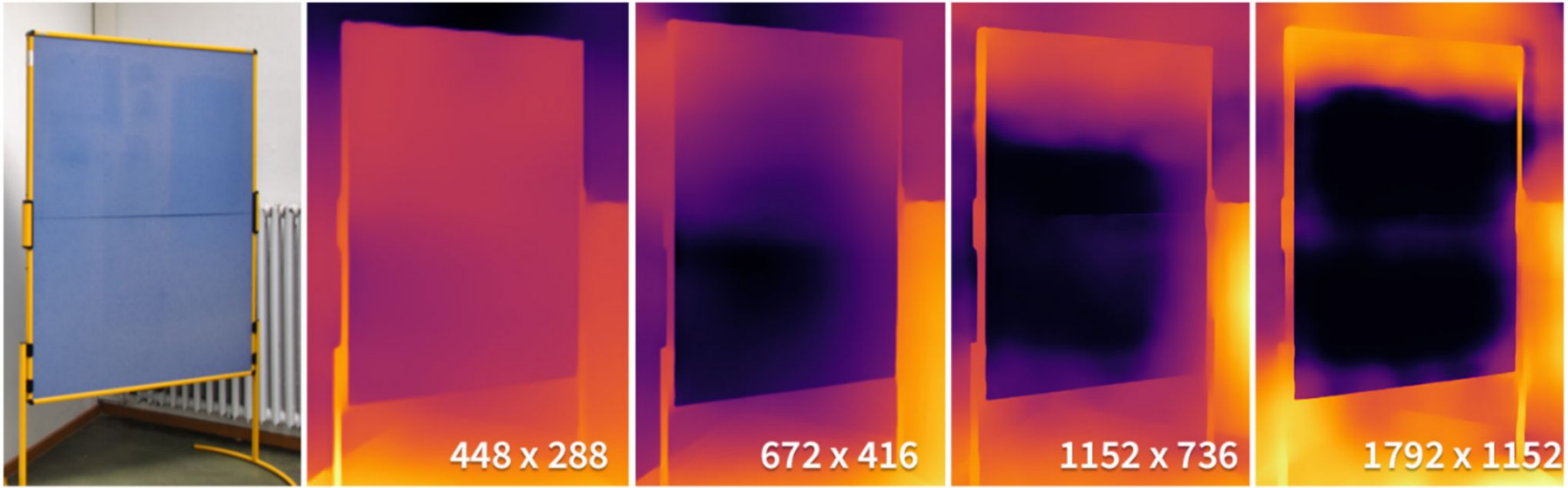




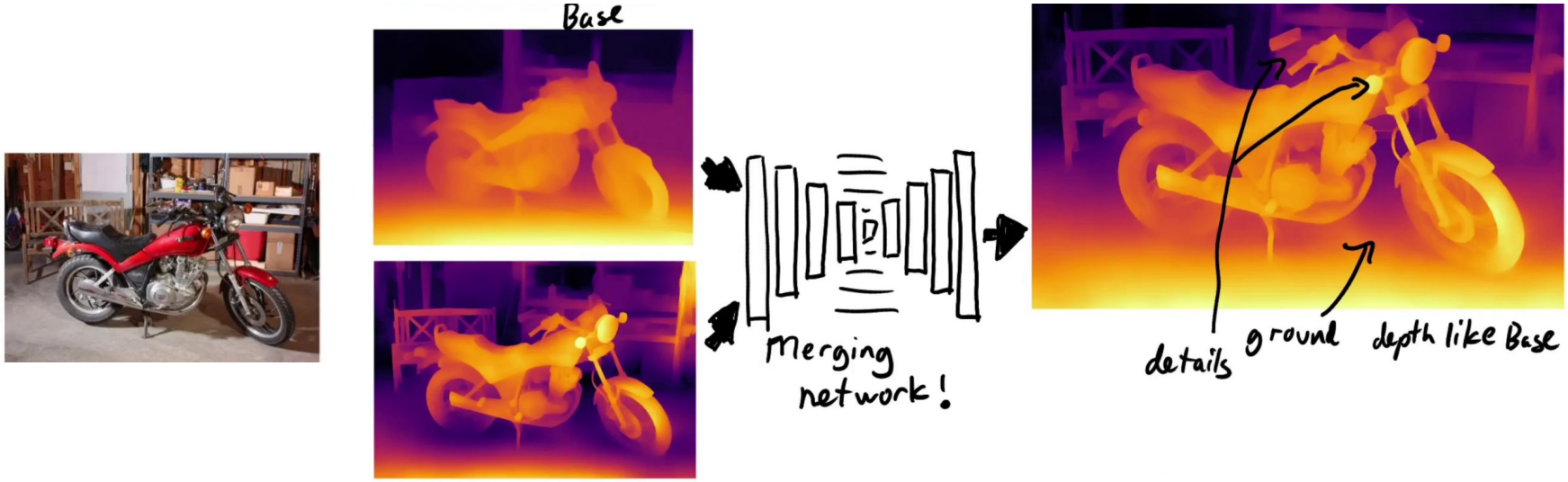
- This is mainly due to the limited capacity and the limited receptive field size of the network.

- As the resolution increases starting from the receptive field size of 448, the network again progressively degrades the accuracy

- The maximum resolution at which the network will be able to generate a consistent structure depends on the distribution of the contextual cues in the image



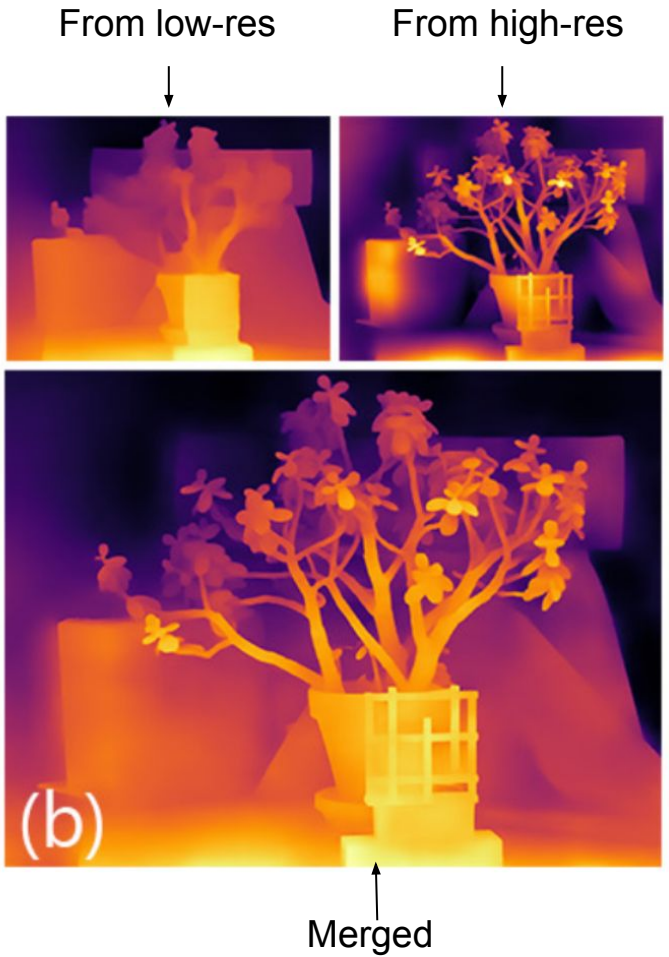
Framework

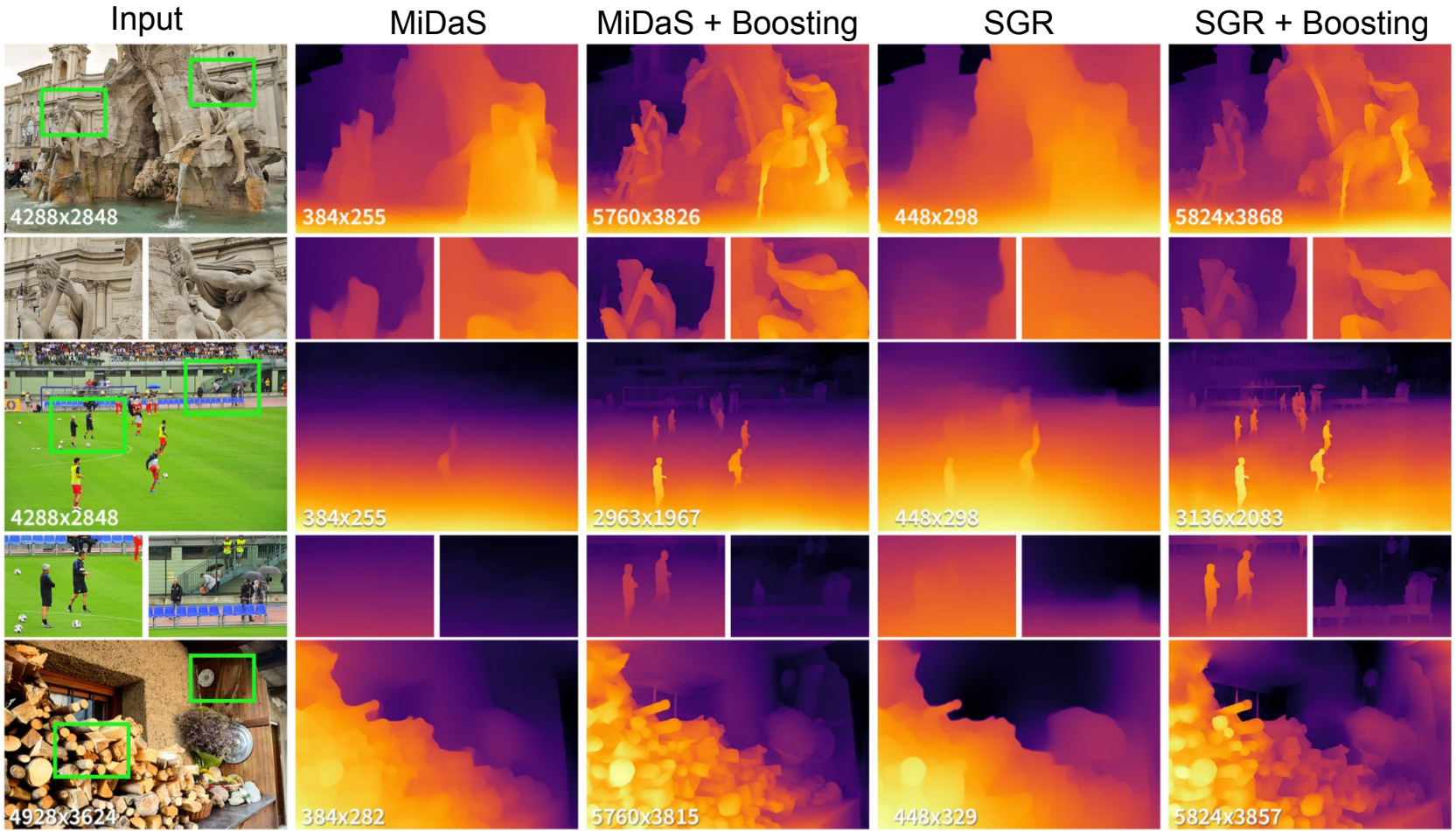


- A **double-estimation framework** that merges two depth estimations for the same image at different resolutions to generate a result with high-frequency details while maintaining the structural consistency

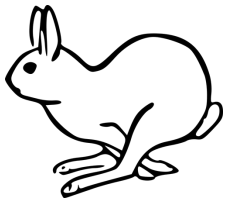
- Generate multi-megapixel depth maps with a high level of detail using a pre-trained model (e.g, MiDAS)


(a)


From low-res
From high-res
(b)
Merged


Input
MiDaS
MiDaS + Boosting
SGR
SGR + Boosting
4288x2848
384x255
5760x3826
448x298
5824x3868
4288x2848
384x255
2963x1967
448x298
3136x2083
4928x3624
384x282
5760x3815
448x329
5824x3857

Can we run such systems everywhere?



High-end GPU (i.e. nVidia Titan X)

Power hungry (250 Watt) - nearly 30 fps (~ 0.035s per frame)

Average CPU (i.e., Intel i7)

Lower energy requirements (~90 Watt)

Less than 2 fps (~ 0.60s per frame)

Embedded CPU (i.e., Raspberry Pi 3)

Extremely low consumption (~3,5 Watt)

Incredibly SLOW (~ 10s per frame)

Towards real-time unsupervised monocular depth estimation on CPU (Poggi, 2018/2022)

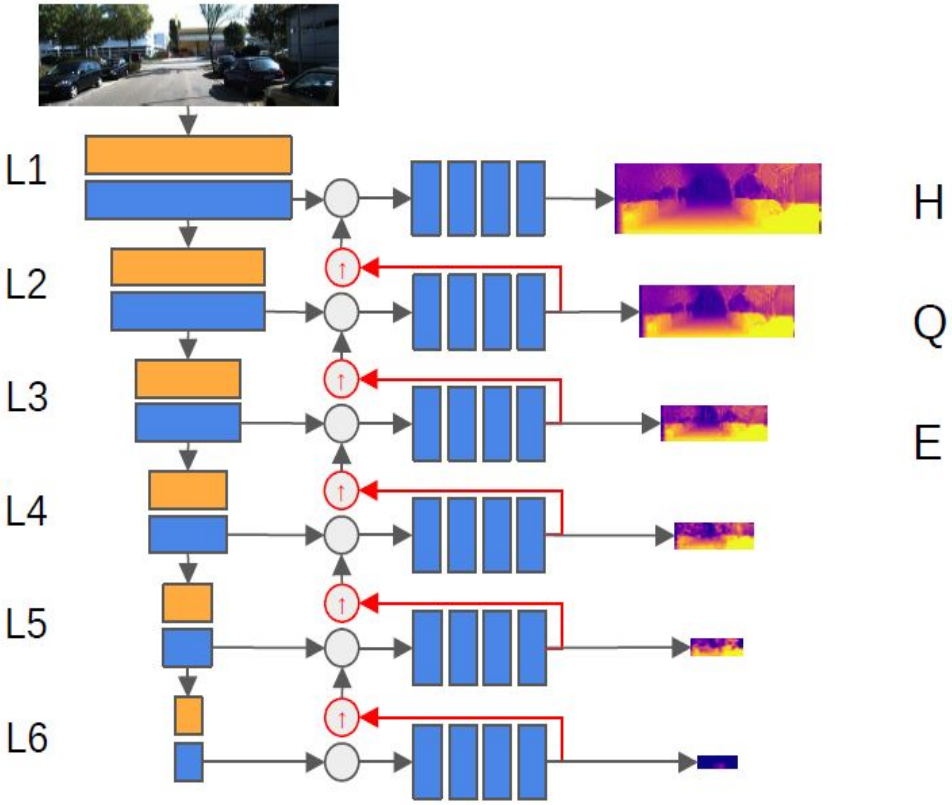
- Current architectures for monocular depth estimation are **very deep** and **complex**; for these reasons they require dedicated hardware such as high-end and power-hungry GPUs.

- This fact precludes to infer depth from a single image in many interesting applications fields characterized by **low-power constraints** (e.g. UAVs, wearable devices, ...)

Code: https://github.com/mattpoggi/pydnet

PyDNet

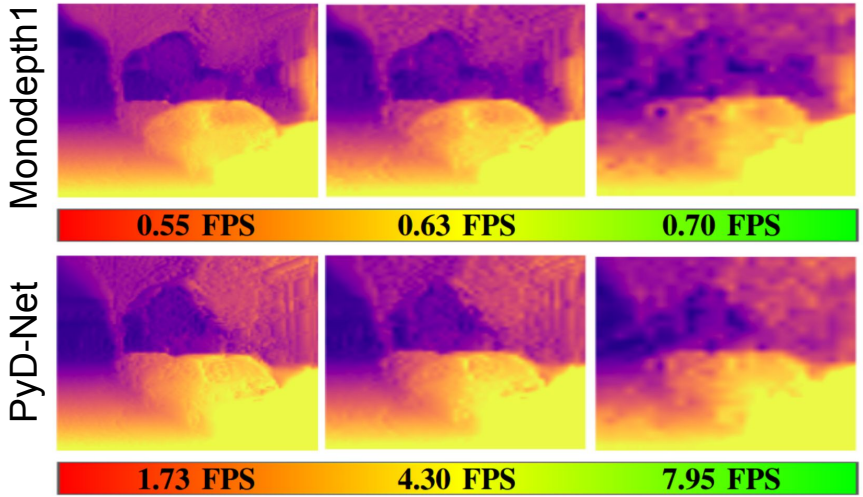
- Shallow, **pyramidal features** encoder
- **Coarse-to-fine** strategy: depth is estimated from lower to higher resolution by lightweight decoders
- Each decoder outputs depth, so as we can **early stop** to trade accuracy for efficiency
- About **6% complexity** compared to Godard et al., CVPR 2017 (1.9M vs 31,6M params
- **Self-supervised** training as Godard et al., CVPR 2017



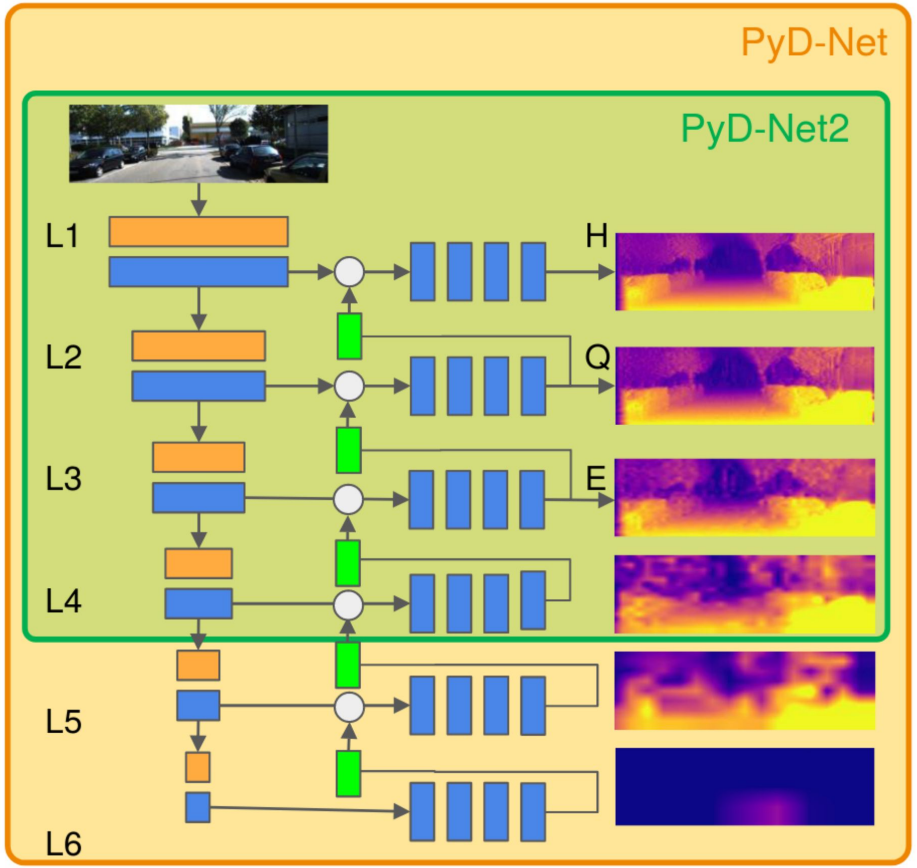
PyDNet

	Power	250+ [W]	91+ [W]	3.5 [W]
Model	Res.	Titan X	i7-6700K	Raspberry Pi 3
Godard et al. [2]	F	0.035 s	0.67 s	10.21 s
Godard et al. [2]	H	0.030 s	0.59 s	8.14 s
PyD-Net	H	0.020 s	0.12 s	1.72 s
Godard et al. [2]	Q	0.028 s	0.54 s	6.72 s
PyD-Net	Q	0.011 s	0.05 s	0.82 s
Godard et al. [2]	E	0.027 s	0.47 s	5.23 s
PyD-Net	E	0.008 s	0.03 s	0.45 s

- Runtime on the ARM Cortex A57 embedded CPU of the NVIDIA Jetson Nano

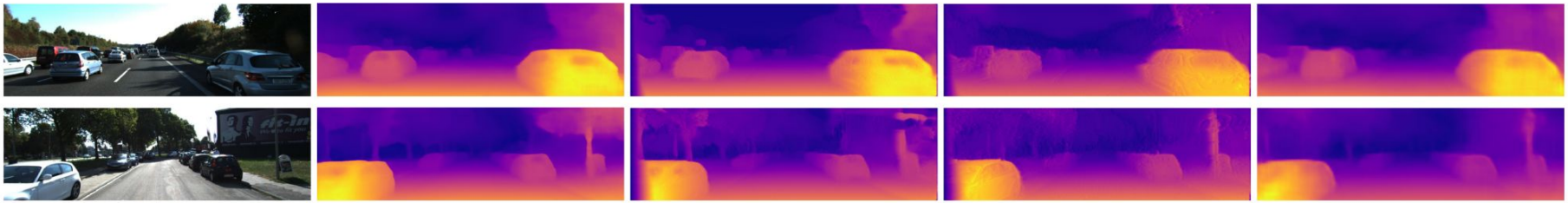


PyDNet2



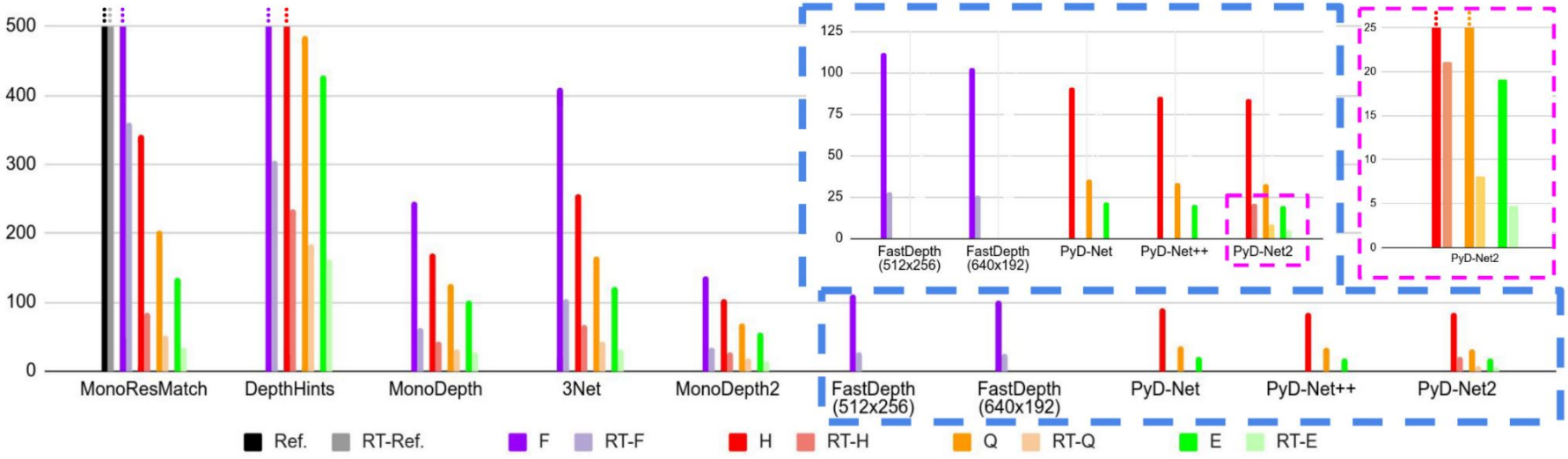
Quantitative results on the KITTI dataset

							Lower is better		Higher is better			
Method	Image Res.	Training	SGM	Pars.	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	FPS
DepthHints [15]	1024×320	I+K	✓	35M	**0.097**	0.733	4.445	**0.186**	**0.889**	**0.962**	**0.981**	36.23
MonoResMatch [14]	1280×384	CS+K	✓	43M	0.098	**0.711**	**4.433**	0.189	0.888	0.960	0.980	12.53
MonoDepth [1]	512×256	CS+K		31M	0.124	1.076	5.311	0.219	0.847	0.942	0.973	77.20
3Net [11]	512×256	CS+K		48M	0.117	0.905	4.982	0.210	0.856	0.948	0.976	56.23
MonoDepth2 [13]	640×192	I+K		15M	0.109	0.873	4.960	0.209	0.864	0.948	0.975	133.51
PyD-Net [2]	512×256	CS+K		1.9M	0.146	1.291	5.907	0.245	0.801	0.926	0.967	203.96
PyD-Net2	640×192	CS+K	✓	**0.7M**	0.127	1.059	5.259	0.218	0.834	0.942	0.974	280.74
PyD-Net2-RT	320×96	CS+K	✓	**0.7M**	0.145	1.260	5.773	0.236	0.797	0.925	0.970	**370.92**



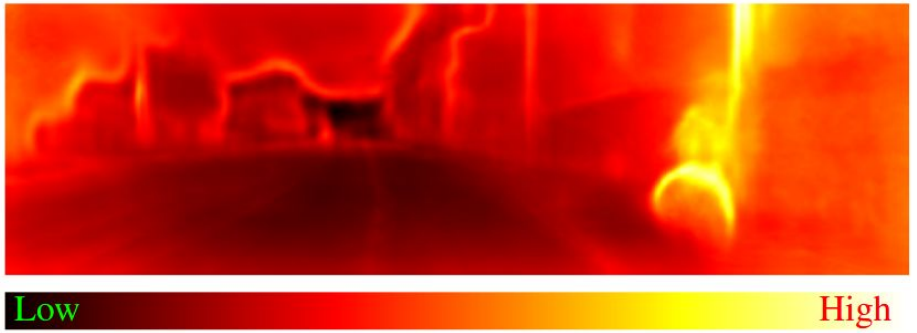
Input image | MonoResMatch [14] | MonoDepth [1] | PyD-Net [2] | PyD-Net2

Memory Footprint



- **RAM usage** for all models, testing on i7 CPU

How much can we trust Self-supervised Monocular Depth Estimation?



On the uncertainty of self-supervised monocular depth estimation (Poggi, 2020)

- As for other perception strategies, it is essential to find out **failure cases** in monocular depth estimation networks
- The erroneous perception of distance to pedestrians or other vehicles might have **dramatic consequences**
- The ill-posed nature of depth-from-mono perception task makes this eventuality much more likely to occur compared to scene geometry (e.g. depth from multiple views)

Depth-from-mono Uncertainties

- **Empirical Estimation**: aims at encoding uncertainty empirically by measuring the variance between a set of all possible network configuration (**epistemic uncertainty**)
  - Dropout Sampling
  - Bootsrapped Ensemble
  - Snapshot Ensemble

- **Predictive Estimation**: these methods produce uncertainty estimates that are function of network parameters and the input image (**aletoric heteroscedastic uncertainty**)
  - Log-Likelihood Maximization
  - Self-Teaching
  - Learning Reprojection

- **Bayesian Estimation**: uncertainty estimates by either placing distributions over model weights, or by learning a direct mapping to probabilistic outputs

Conclusion and Discussion

- **Deep learning** based methods (CNNs) demonstrated a strong ability to accurately estimate dense depth maps from a single image
- **Self-supervised** methodologies to overcome the lack of ground truth depth data (stereo or videos at training time)
- Depth known up to a **scale factor** (except some situations, e.g. stereo)
- Although the great advances in this field, mono solutions are much **less reliable** than stereo/multi-view stereo approach (no geometry)
- **High-resolution** estimation is still an open-problem but great advances for **real-time performances** suited for many applications
- The problem of the **domain shift** is even more evident w.r.t deep learning based solutions for stereo