Single-View Depth Image Estimation

Fangchang Ma
PhD Candidate at MIT (Sertac Karaman Group)

- homepage: www.mit.edu/~fcma/
- code: github.com/fangchangma

MIT
AEROASTRO
LIDS LABORATORY FOR INFORMATION & DECISION SYSTEMS

Depth sensing is key to robotics advancement

1979, Multi-view vision and the Stanford Cart

Depth sensing is key to robotics advancement

2007, Velodyne LiDAR and the DARPA Urban Challenge

Depth sensing is key to robotics advancement

2010, Kinect and aggressive drone manuvers

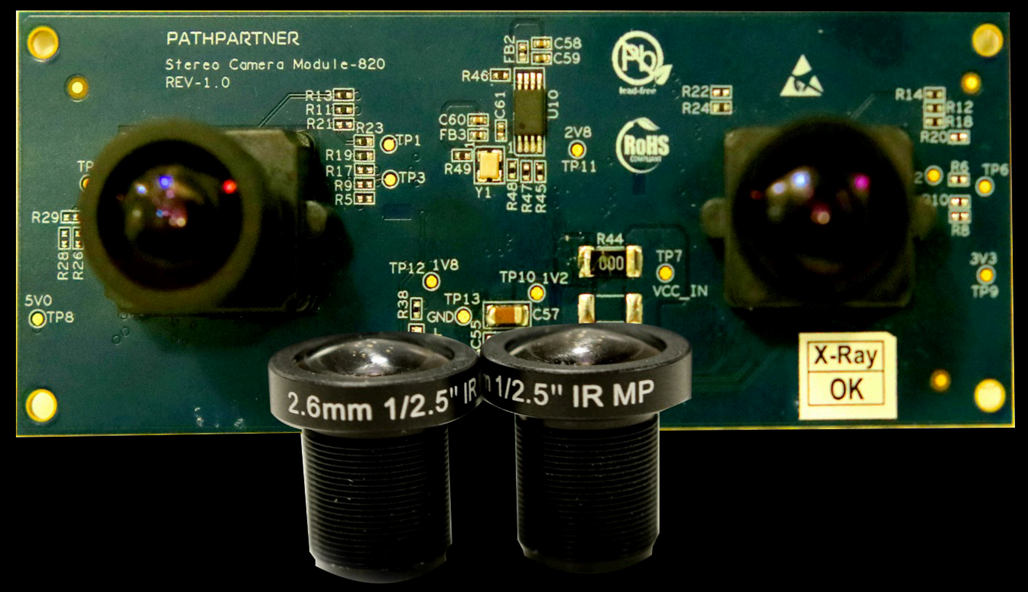
Impact of depth sensing beyond robotics

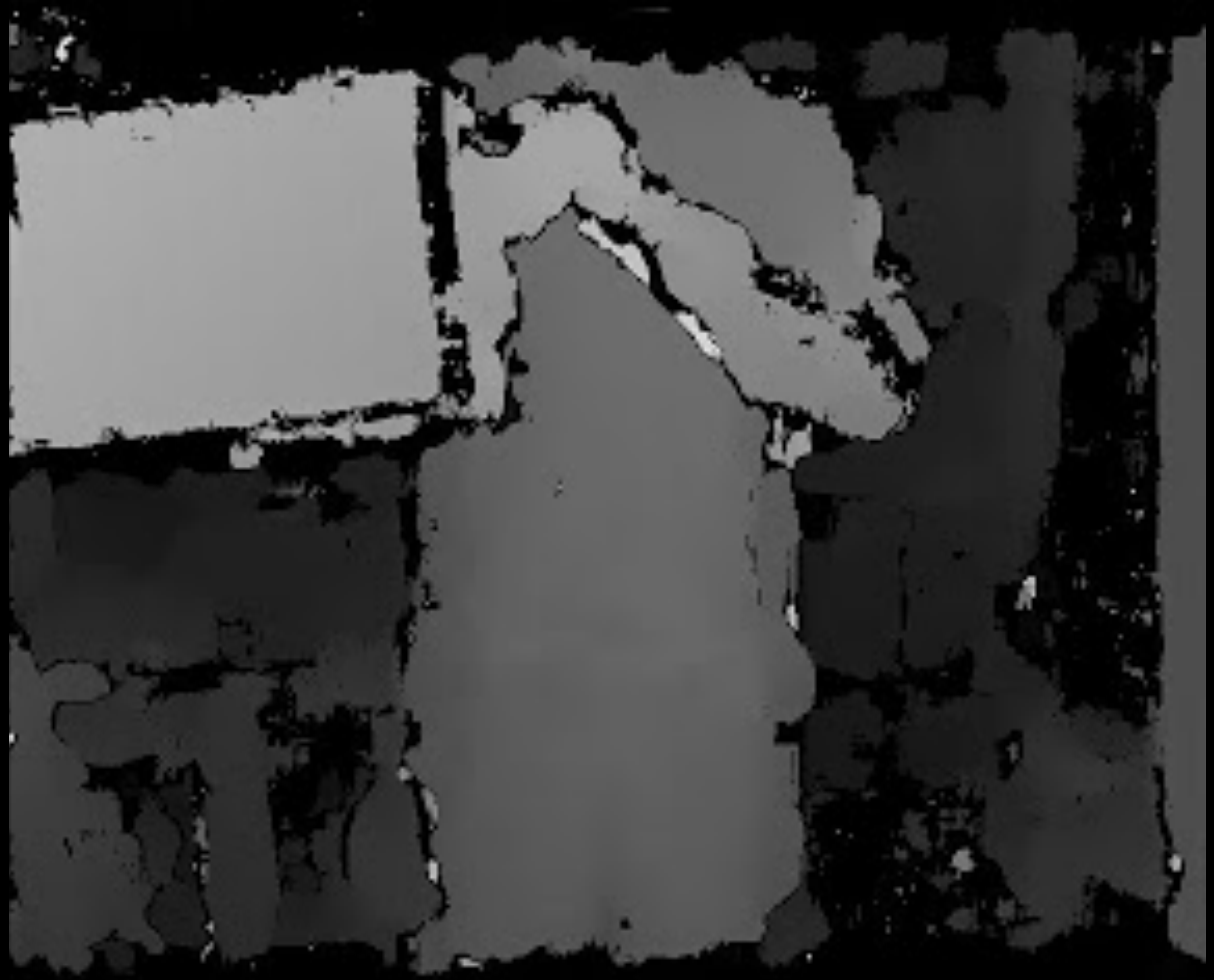
Face ID by Apple

Existing depth sensors have limited effective spatial resolutions

- Stereo Cameras
- Structure-light sensors
- Time-of-flight sensors (e.g., LiDARs)

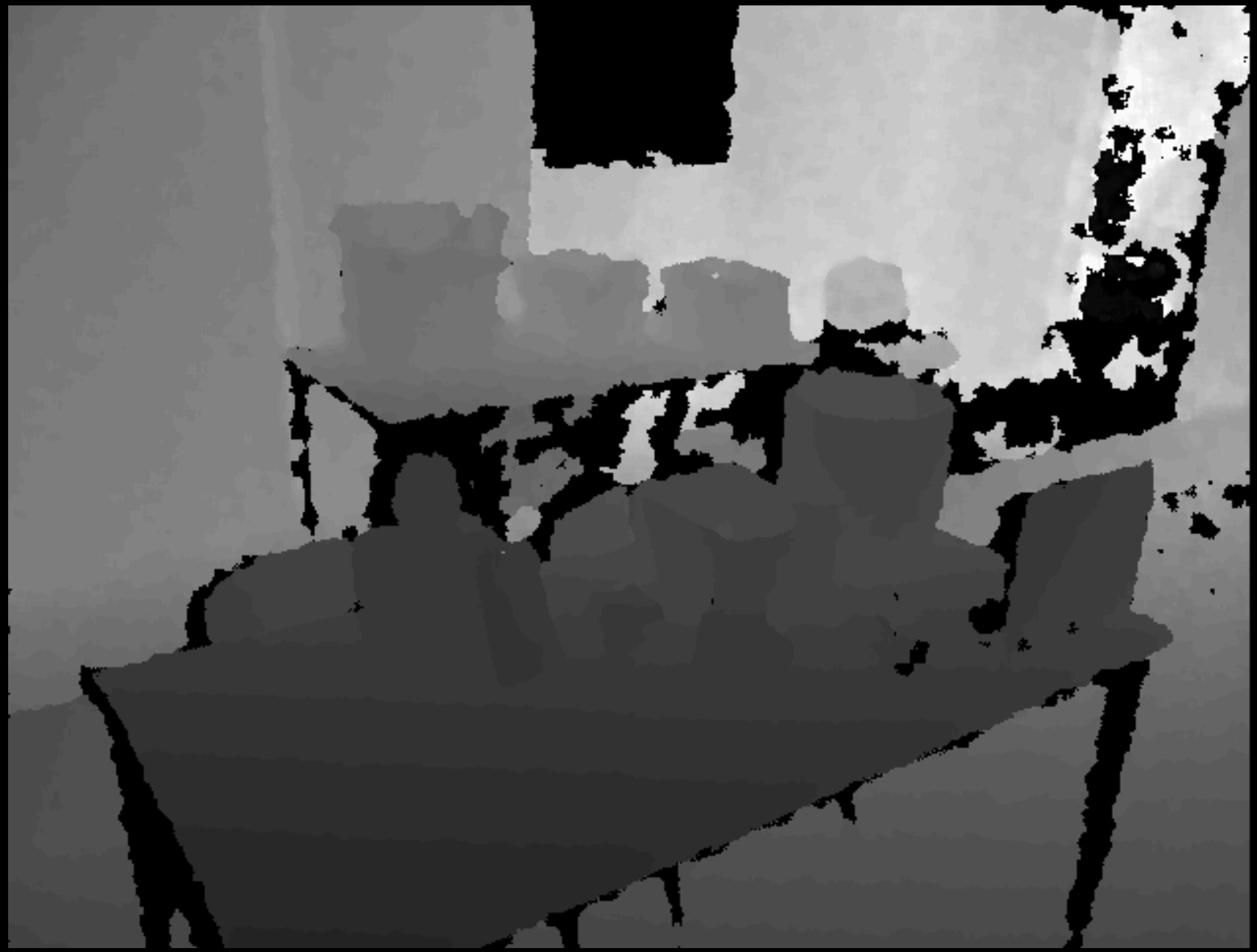
Existing depth sensors have limited effective spatial resolutions





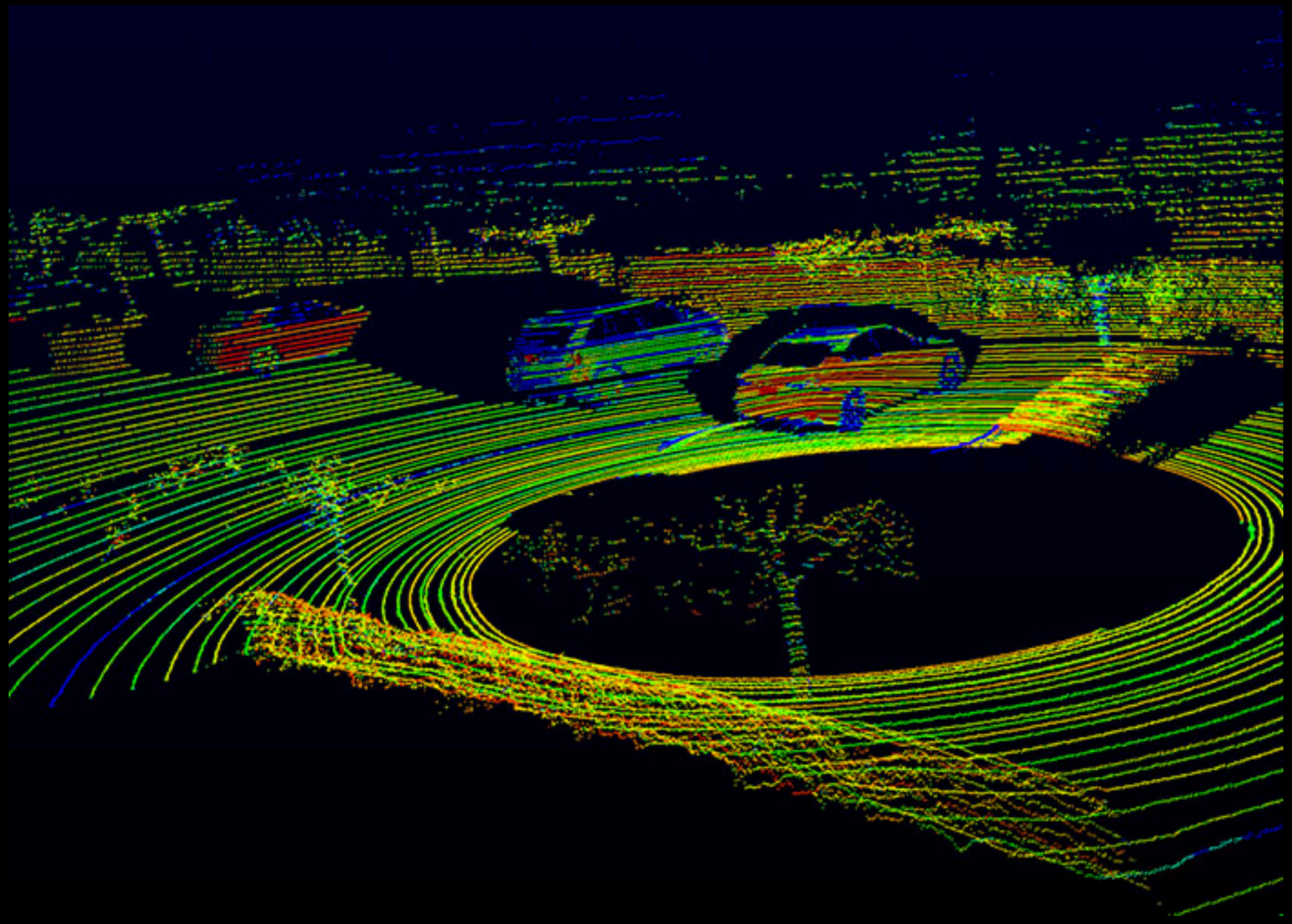
Stereo: triangulation is accurate only at texture-rich regions

Existing depth sensors have limited effective spatial resolutions



Structure-light Sensors: short range, high power consumption

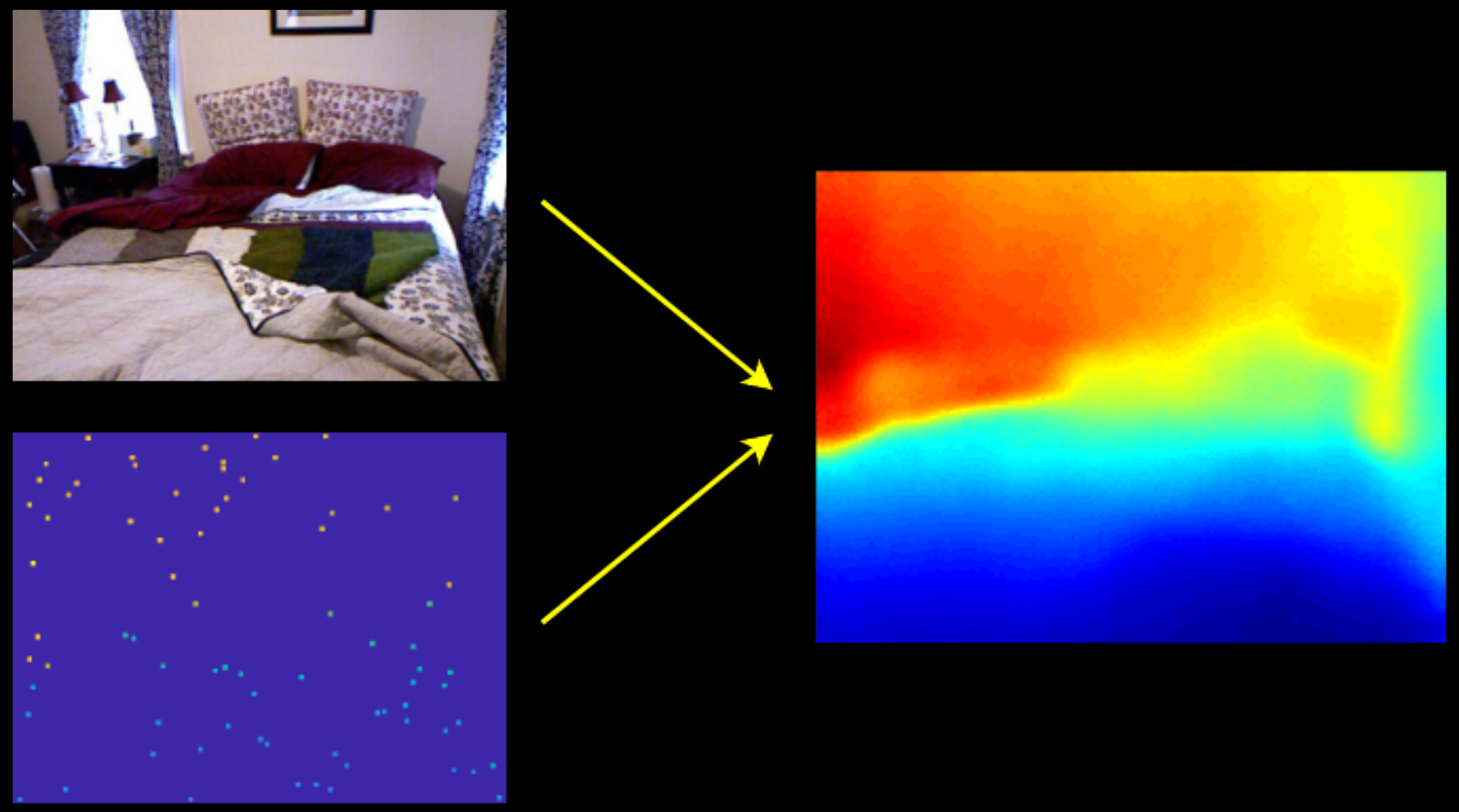
Existing depth sensors have limited effective spatial resolutions



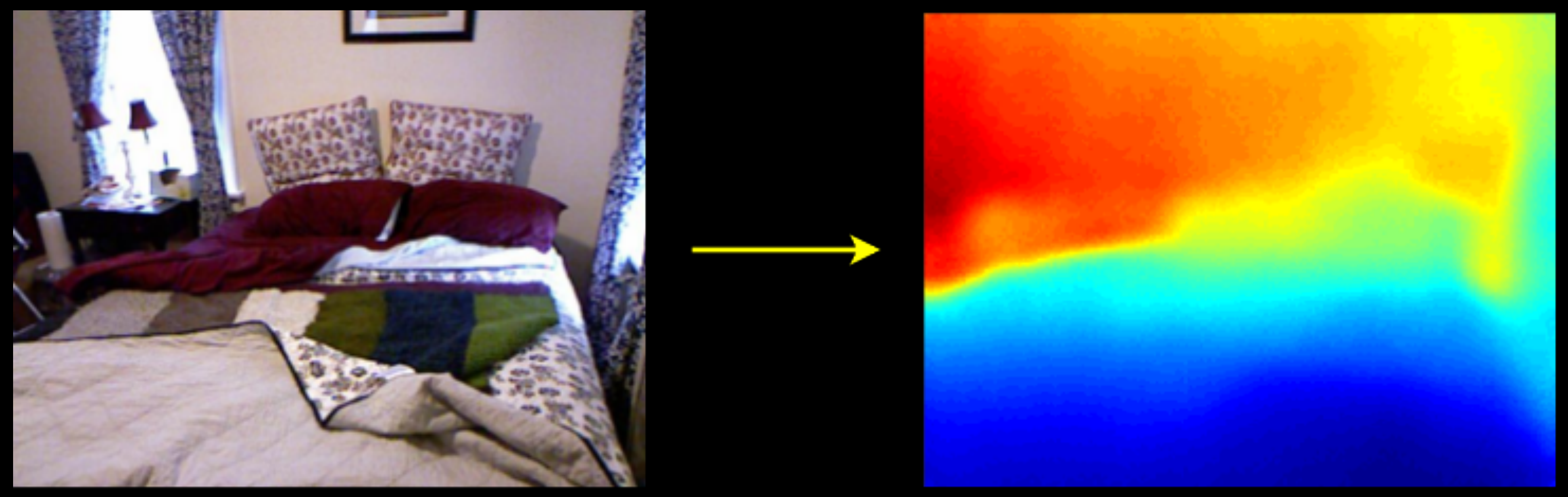
LiDARs: extremely sparse measurements

Single-View Depth Image Estimation

Depth completion



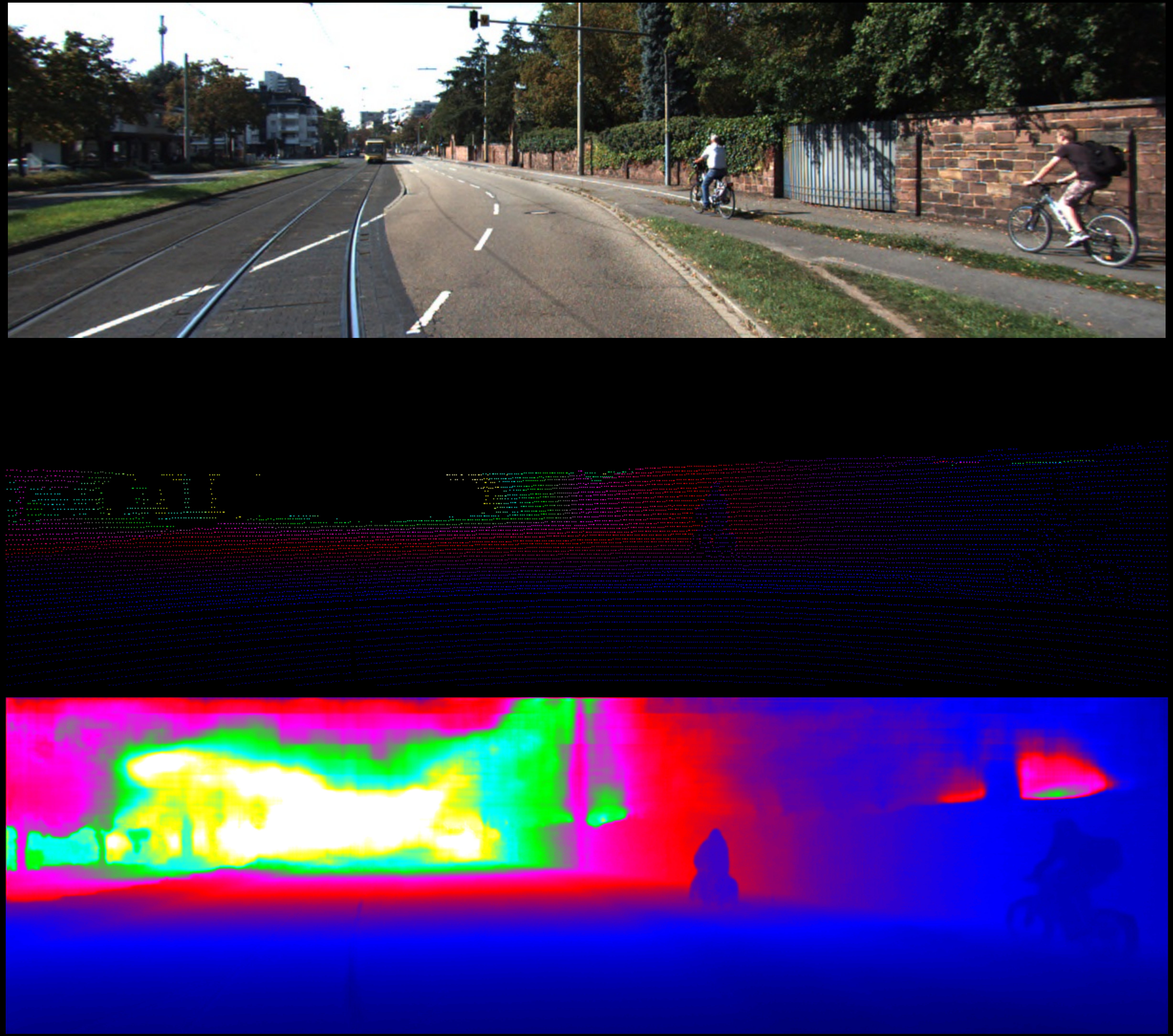
Depth Prediction



Application 1: Sensor Enhancement

Kinect

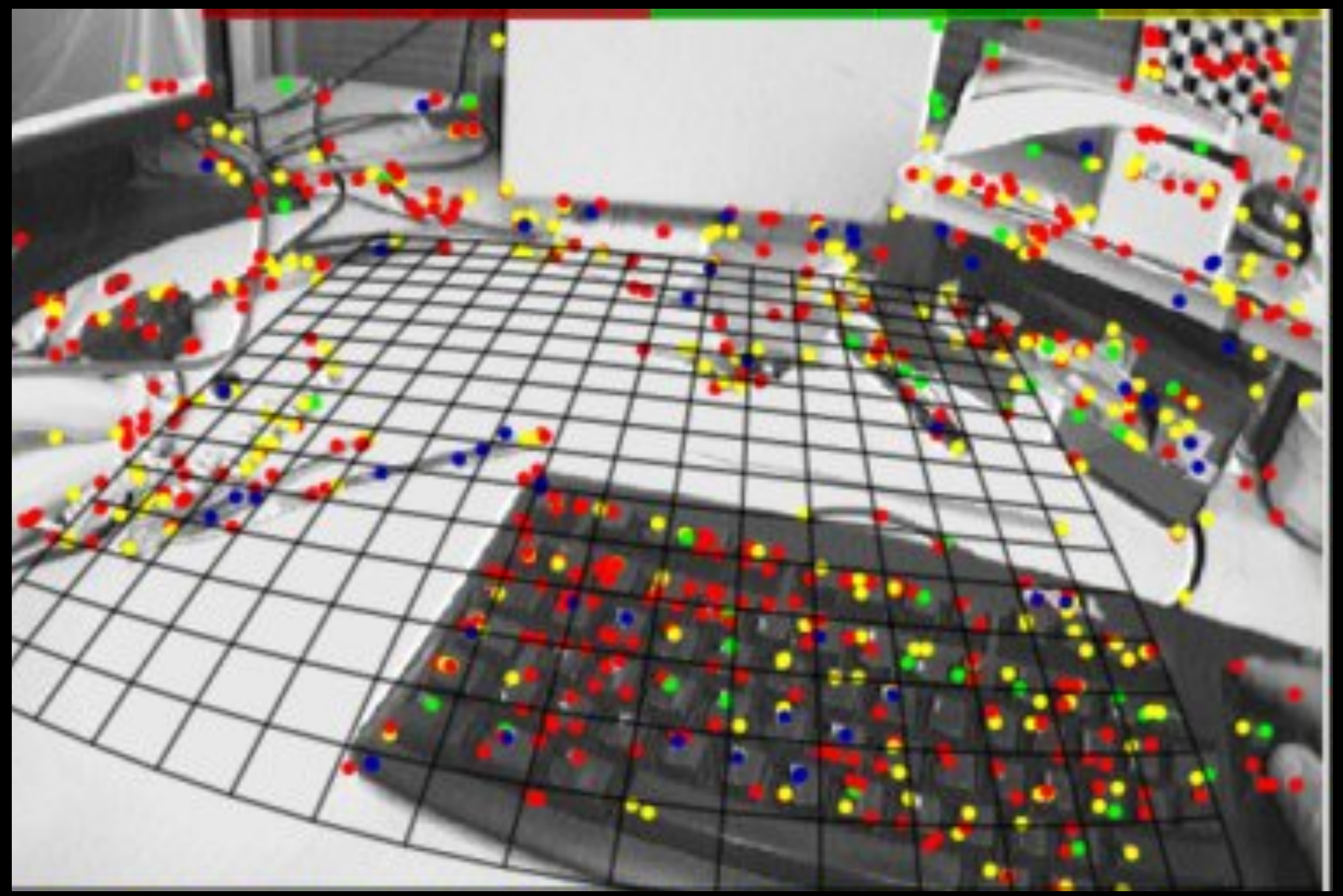
Velodyne LiDAR



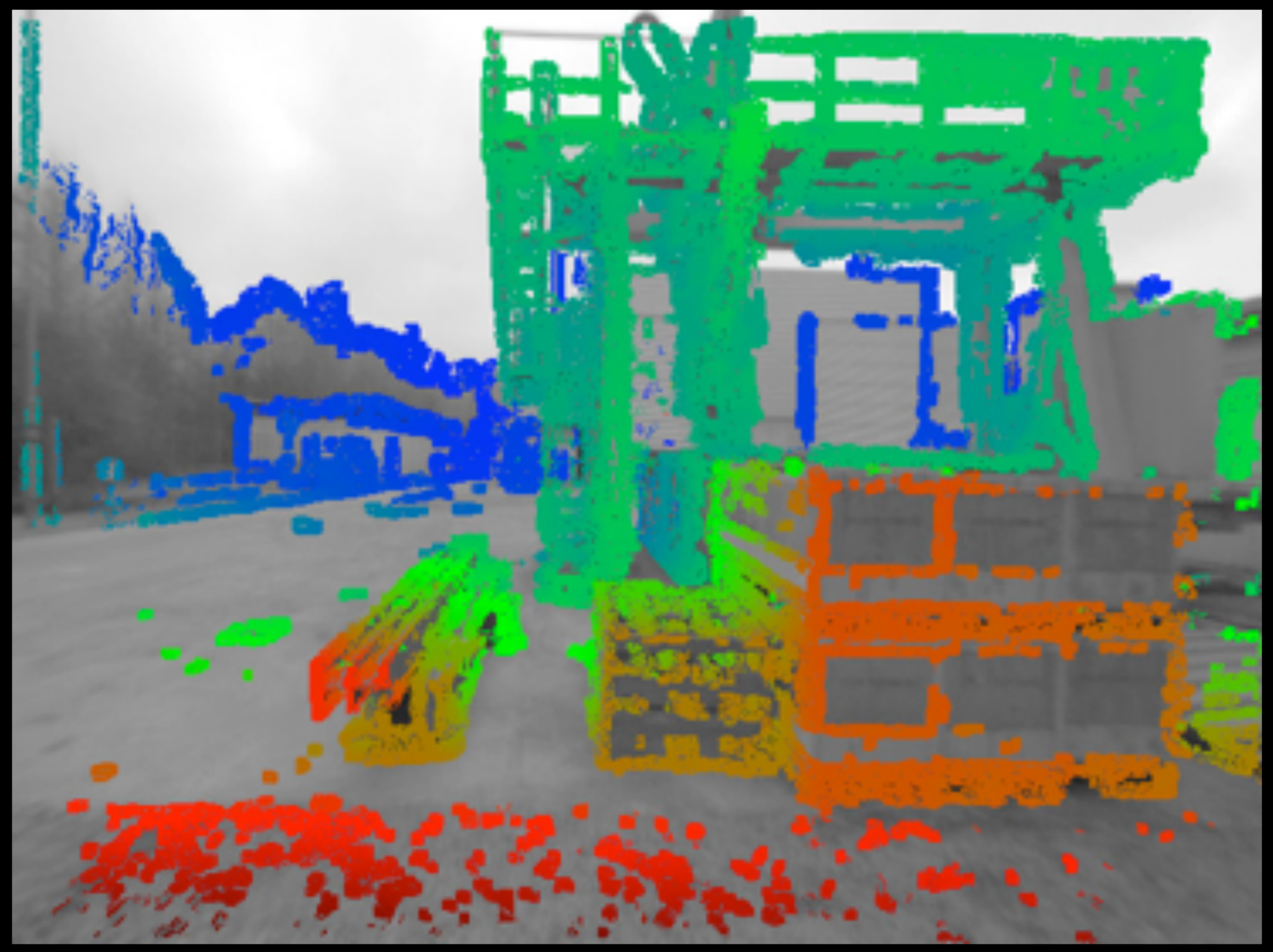
Application 2: Sparse Map Densification

State-of-the-art, real-time SLAM algorithms are mostly (semi) feature-based, resulting in a sparse map representation

PTAM



LSD-SLAM



Depth completion as a downstream, post-processing step for sparse SLAM algorithms, creating a dense map representation

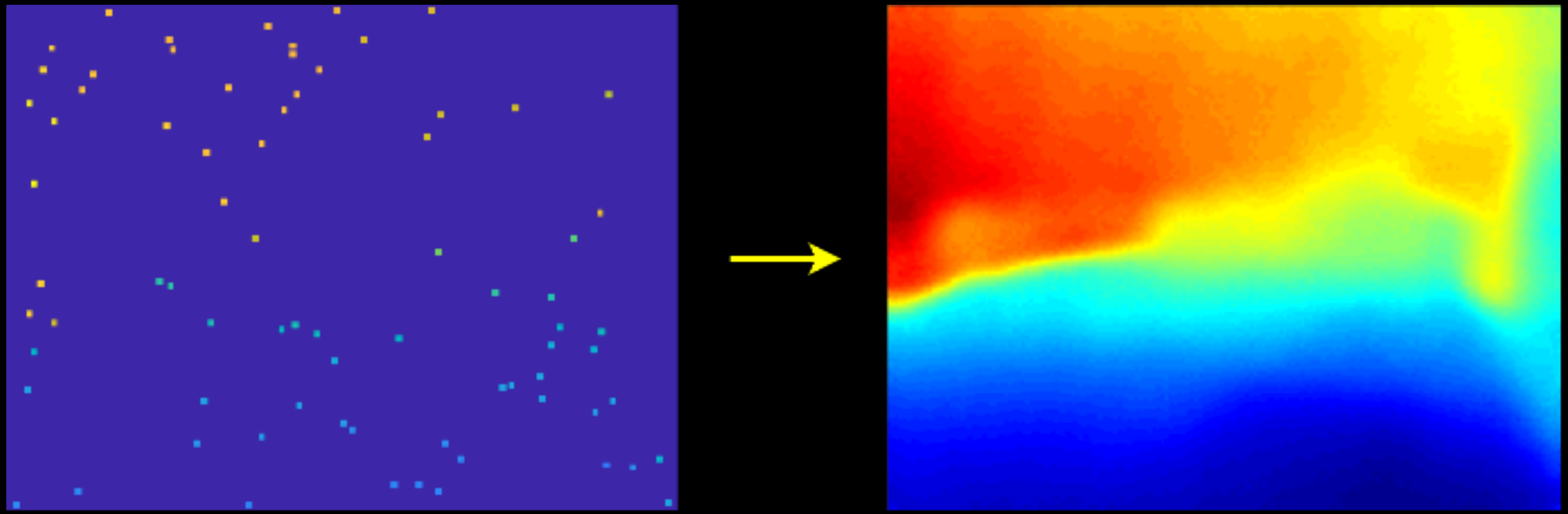
Single-View Depth Image Estimation

- Why is the problem challenging?
- How to solve the problem?
- How to train a model without ground truth?
- How fast can we run on embedded systems?
- How to obtain performance guarantees with DL?
- What to do if you “hate” deep learning?

Single-View Depth Image Estimation

- Why is the problem challenging?
- How to solve the problem?
- How to train a model without ground truth?
- How fast can we run on embedded systems?
- How to obtain performance guarantees with DL?
- What to do if you “hate” deep learning?

Challenges in Depth Completion

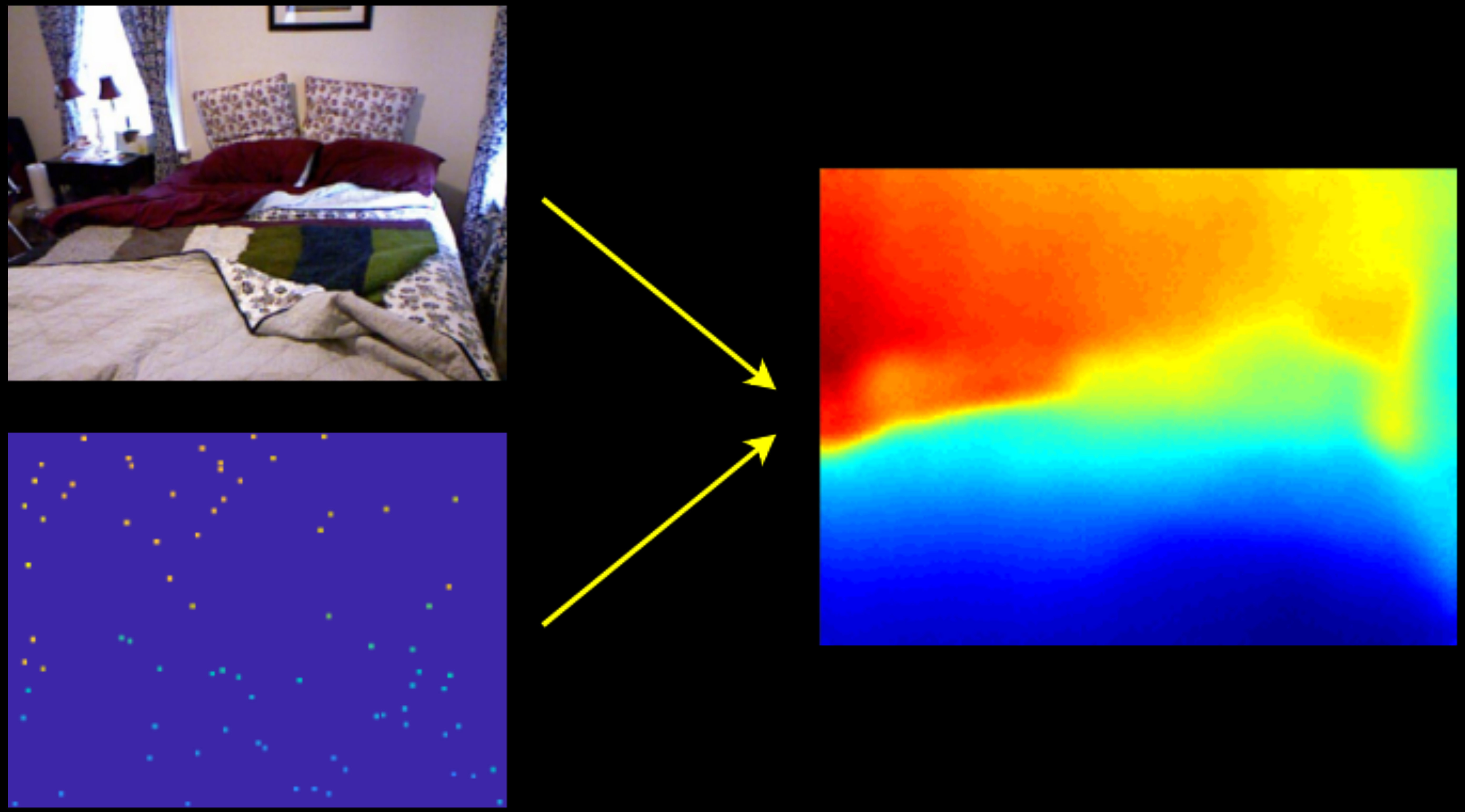


- An ill-posed inverse problem
- High-dimensional, continuous prediction

Challenges in Depth Completion

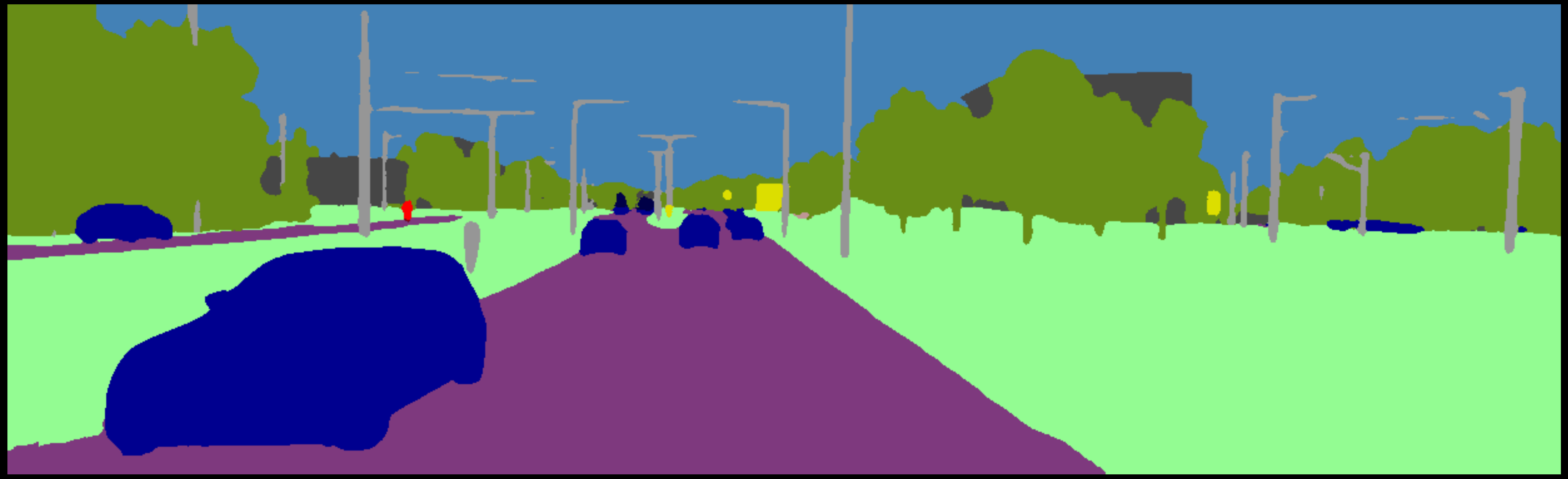
- Biased / adversarial sampling
- Varying number of measurements

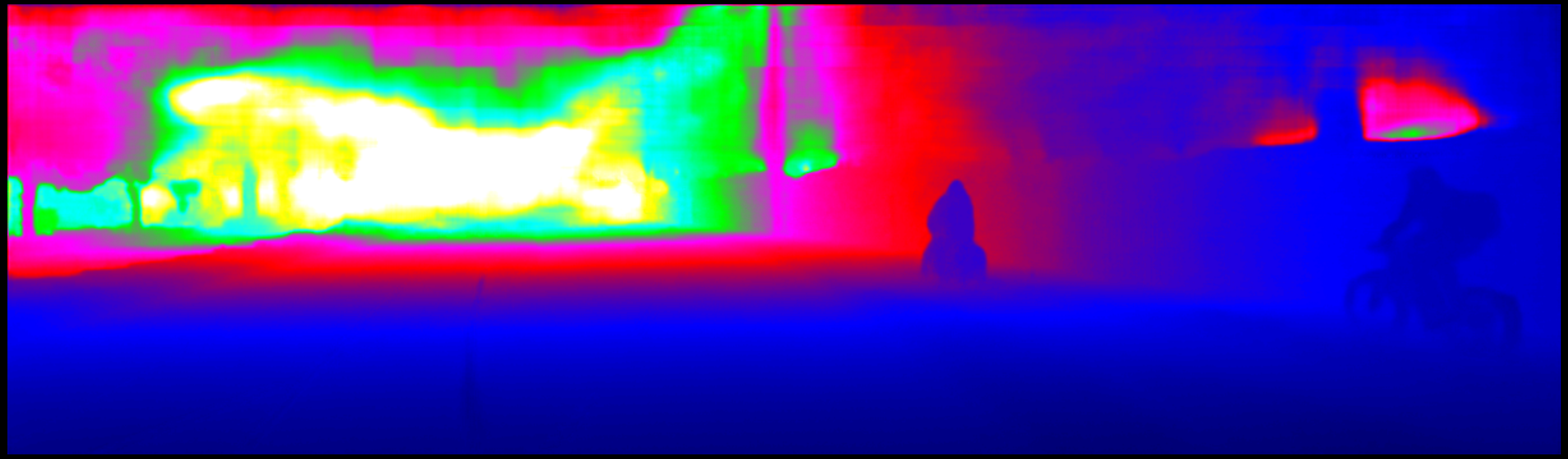
Challenges in Depth Completion



- Cross-modality fusion (RGB + Depth)

Challenges in Depth Completion



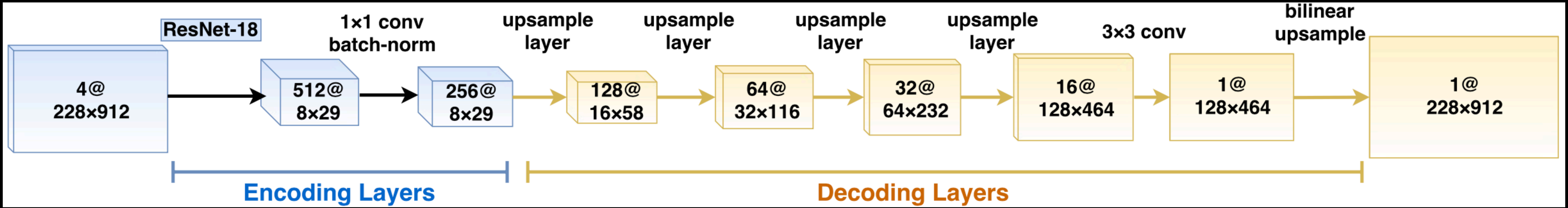


- Lack of ground truth data (category vs. distance)

Single-View Depth Image Estimation

- Why is the problem challenging?
- How to solve the problem?
- How to train a model without ground truth?
- How fast can we run on embedded systems?
- How to obtain performance guarantees with DL?
- What to do if you “hate” deep learning?

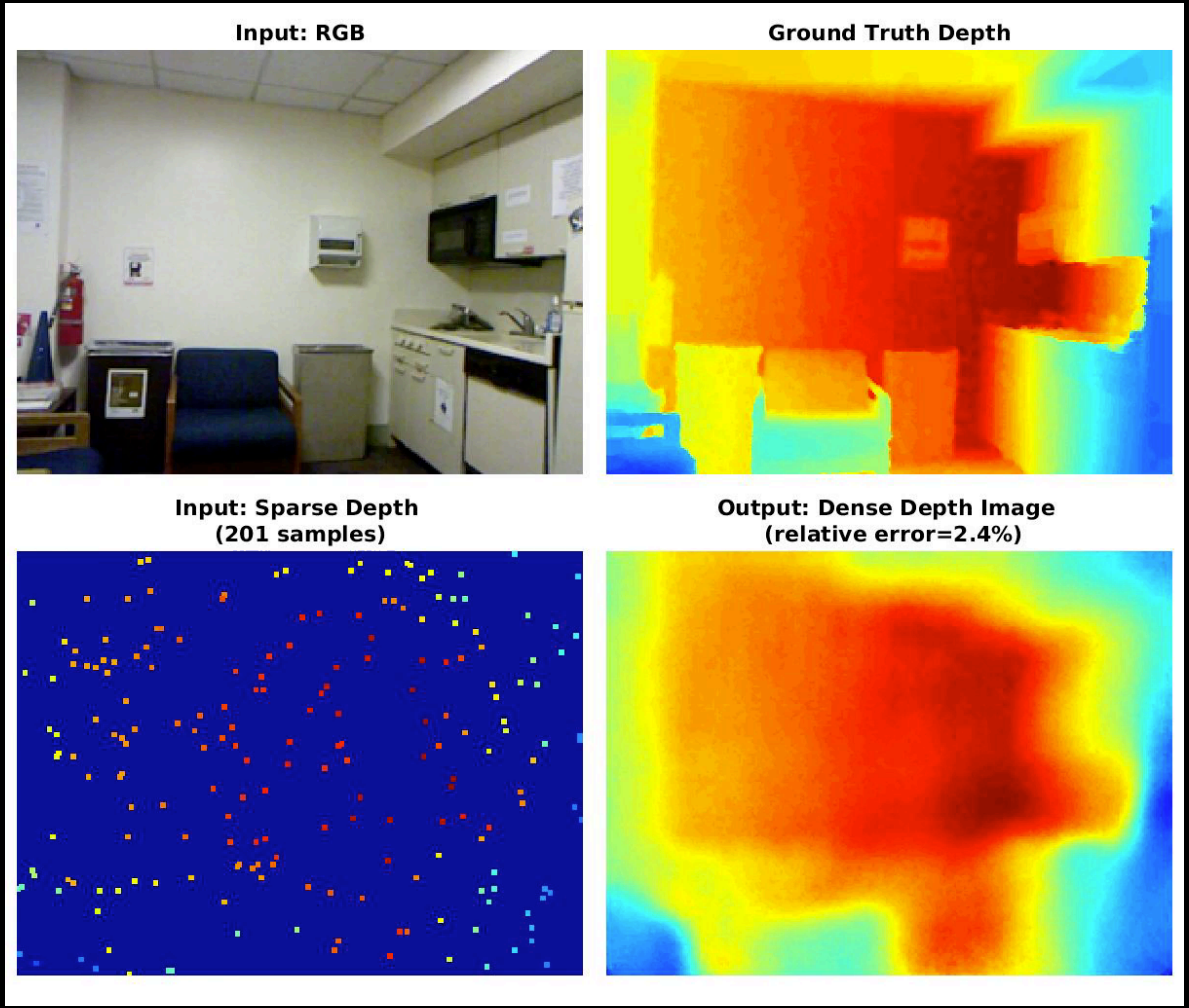
Sparse-to-Dense: Deep Regression Neural Networks



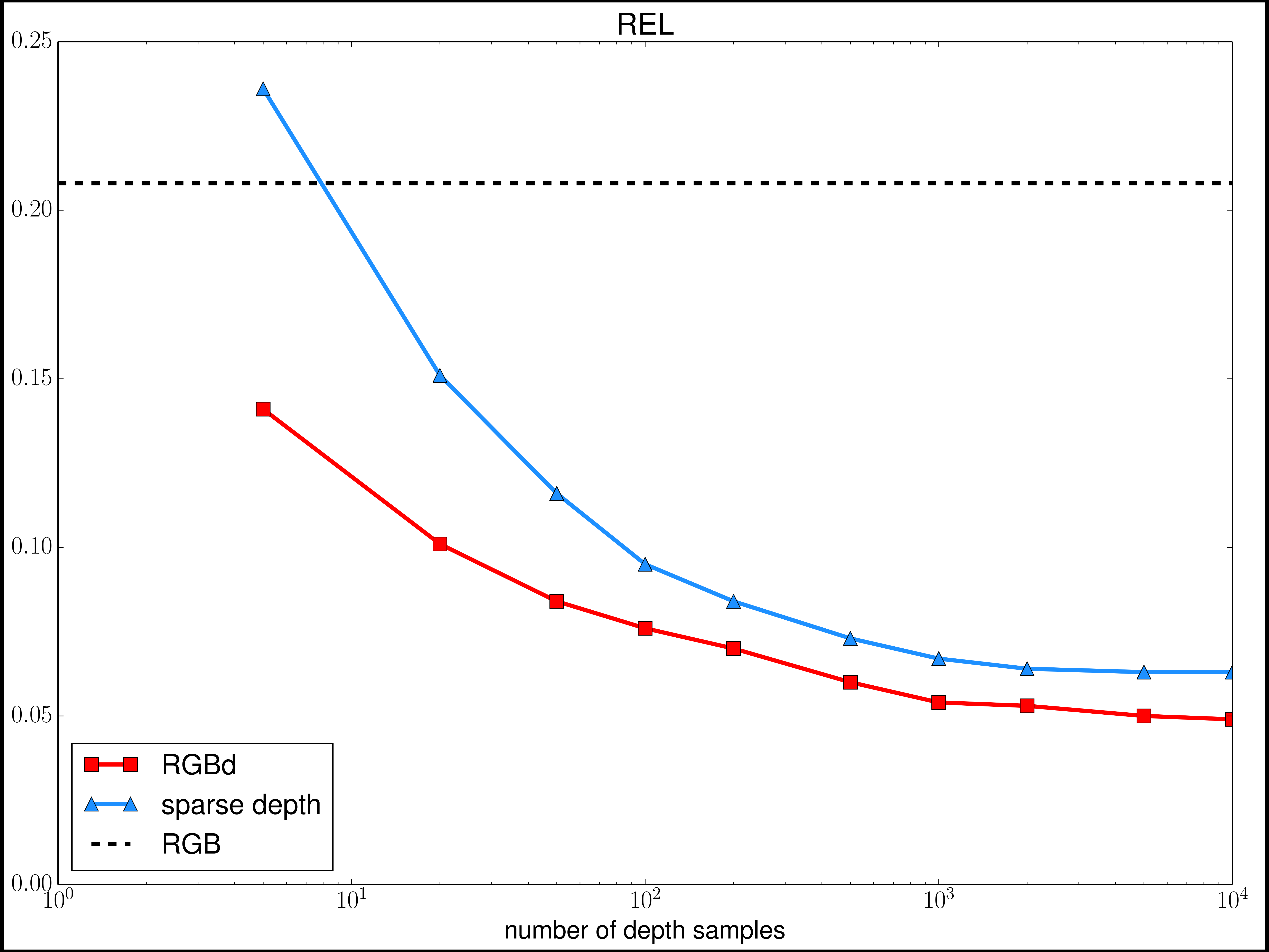
- Direct encoding: use 0s to represent no-measurement
- Early-fusion strategy: concatenate RGB and sparse Depth at input level
- Network Architecture: standard convolutional neural network
- Train end-to-end using ground truth depth

Results on NYU Dataset

- RGB only: RMS=51cm
- RGB + 20 measurements: RMS=35cm
- RGB + 50 measurements: RMS=28cm
- RGB + 200 measurements: RMS=23cm

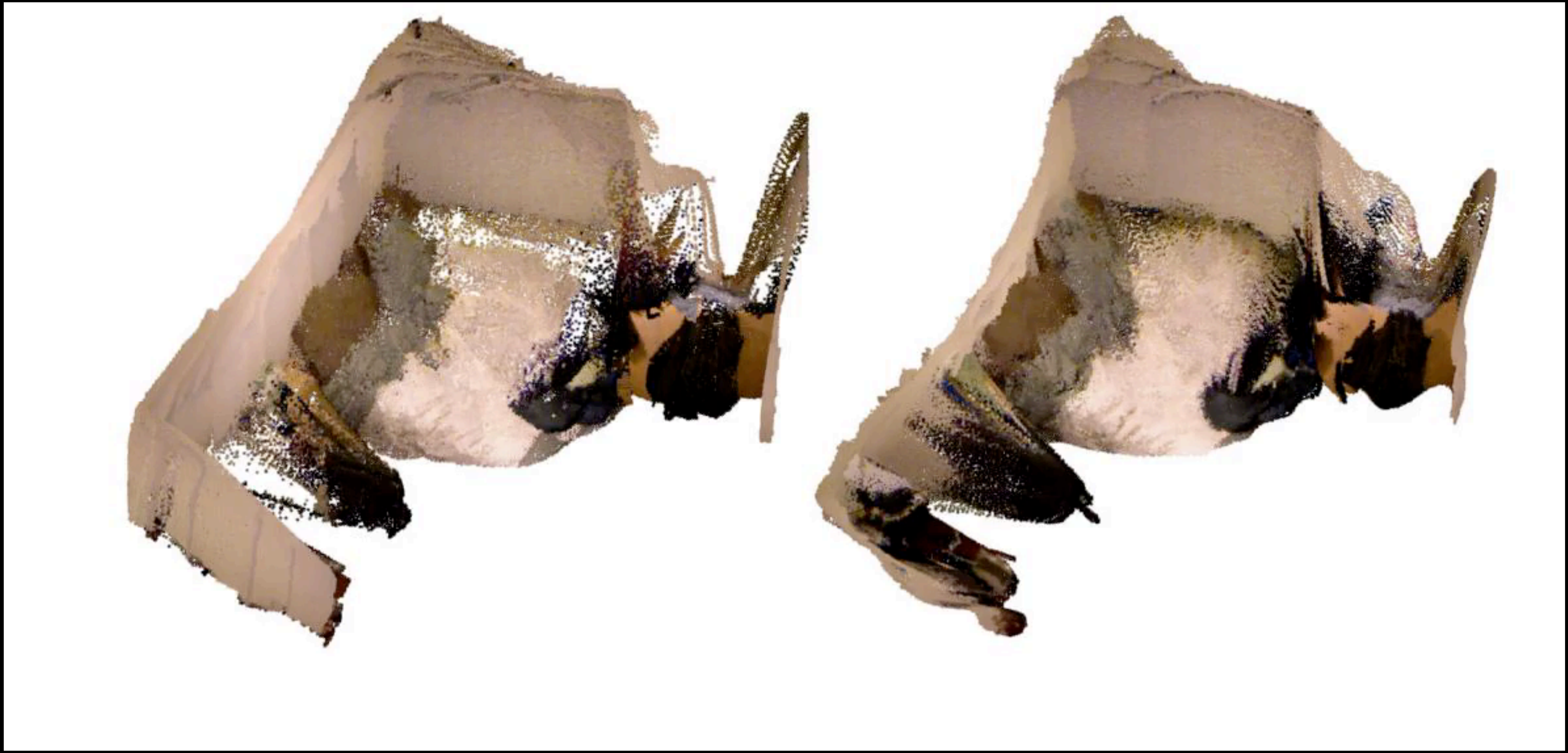


Scaling of Accuracy vs. Samples



Application to Sparse Point Clouds

Application to Sparse Point Clouds



Sparse-to-Dense:
depth prediction from sparse depth samples and a single image

Fangchang Ma, Sertac Karaman

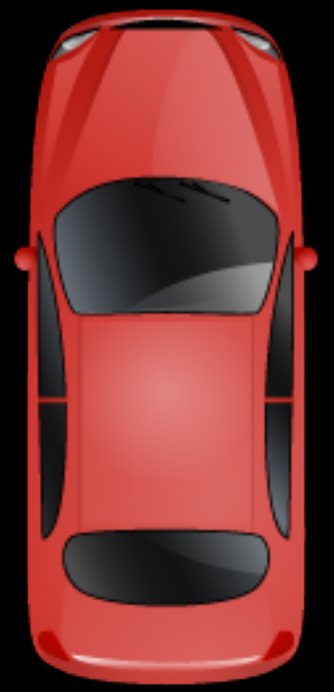
ICRA'18
code: github.com/fangchangma/sparse-to-dense

Single-View Depth Image Estimation

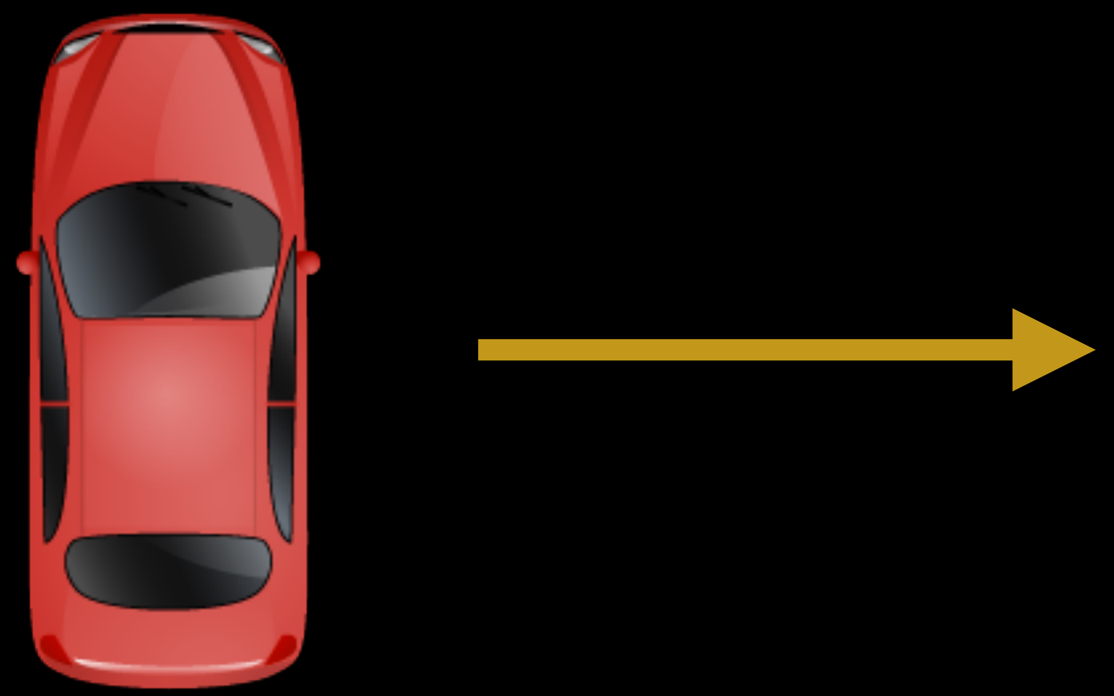
- Why is the problem challenging?
- How to solve the problem?
- How to train a model without ground truth?
- How fast can we run on embedded systems?
- How to obtain performance guarantees with DL?
- What to do if you “hate” deep learning?

Experiment 1. Supervised Training (Baseline). RMSE=0.814m (ranked 1st on KITTI).

Self-supervision: enforce temporal photometric consistency

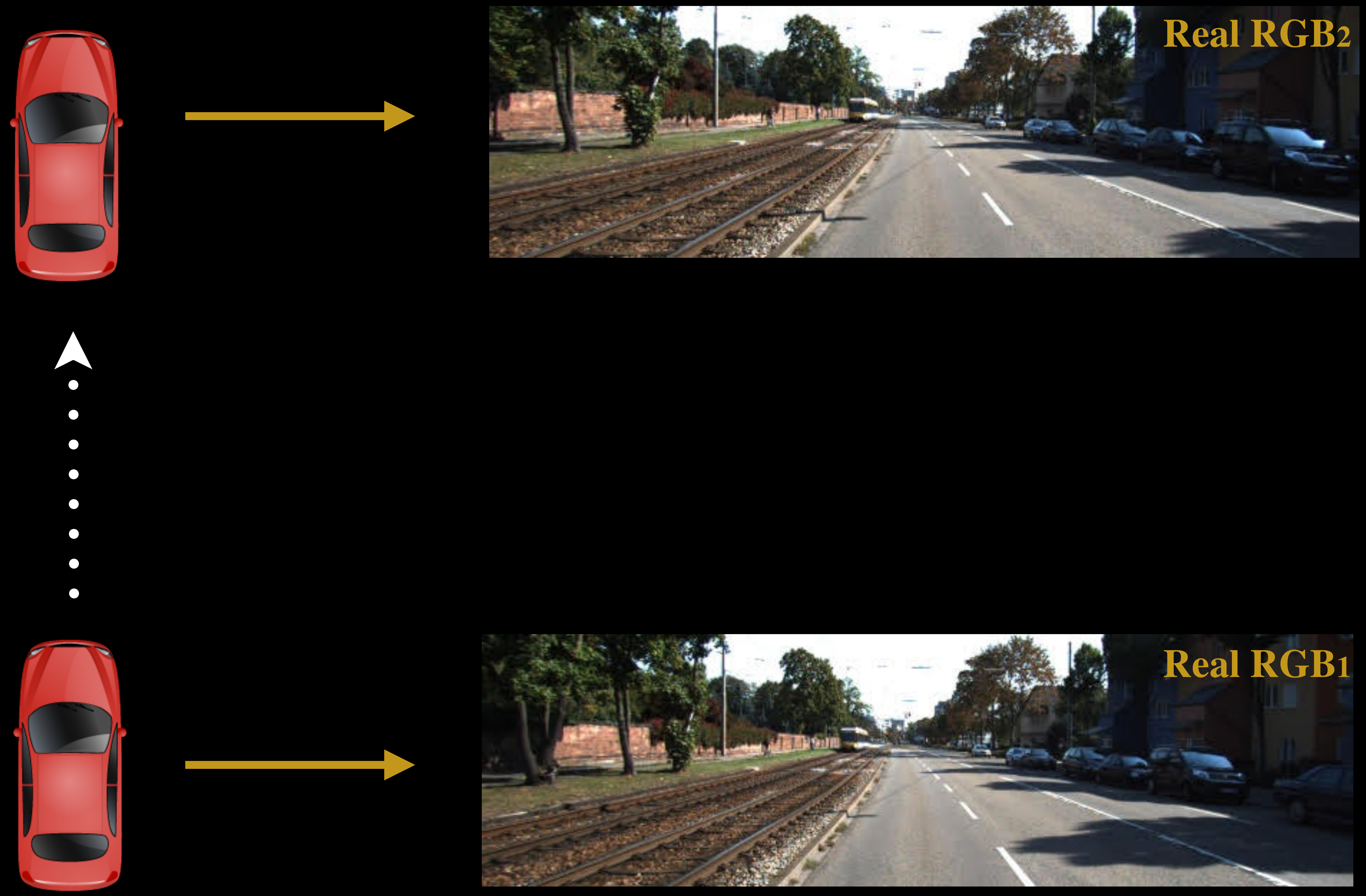


Self-supervision: enforce temporal photometric consistency



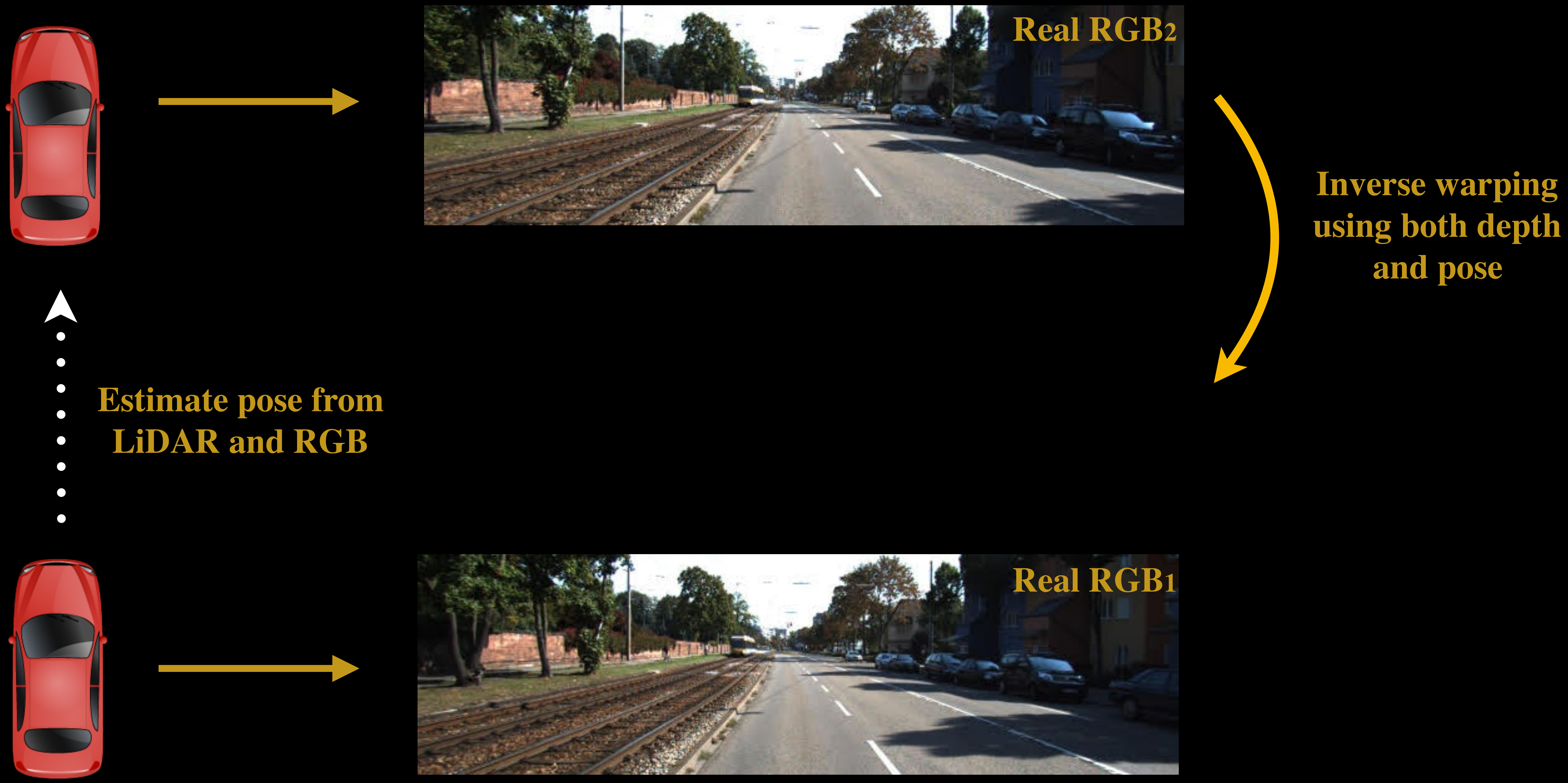
Self-supervision: enforce temporal photometric consistency

Self-supervision: enforce temporal photometric consistency

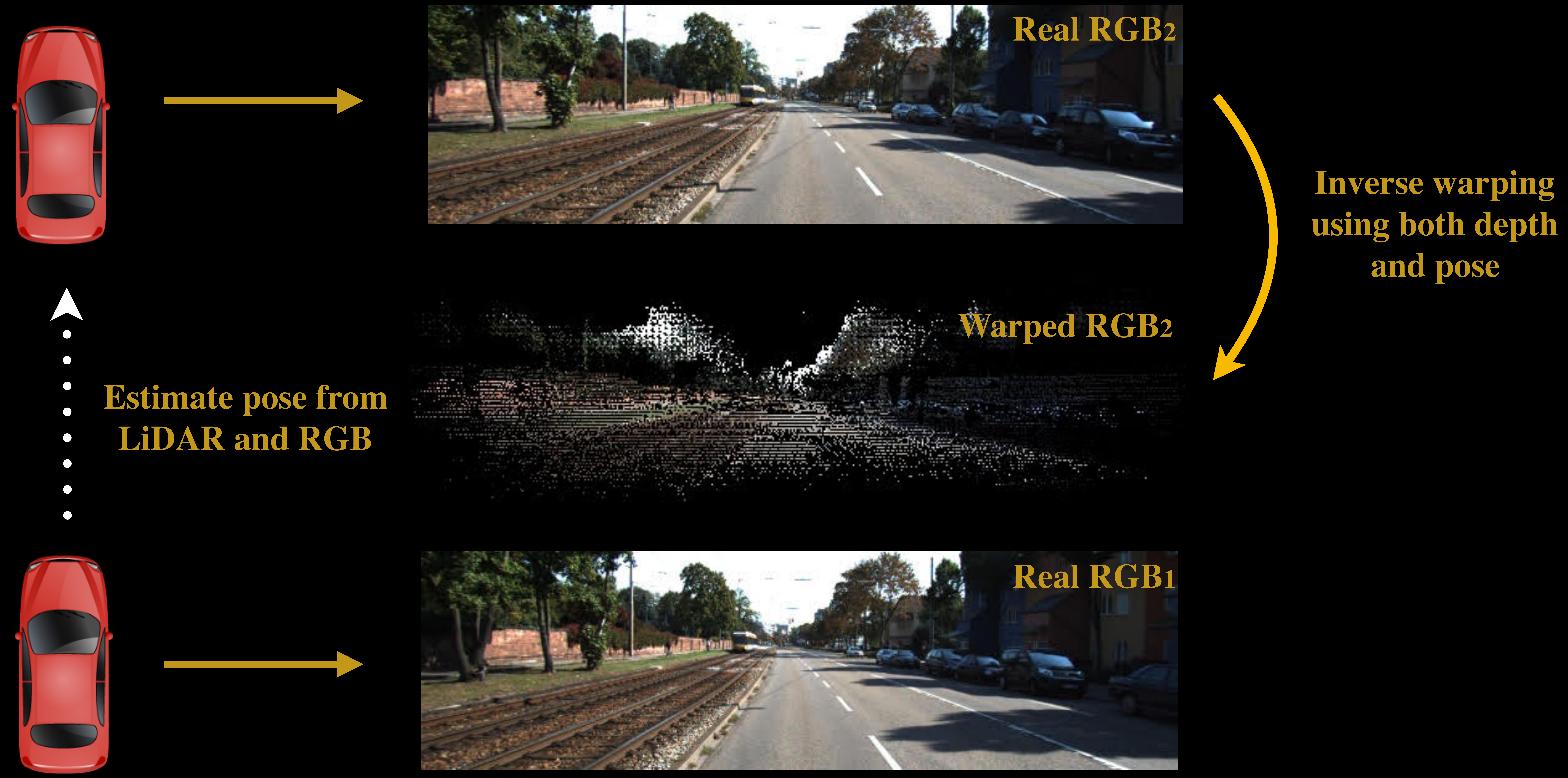


Self-supervision: enforce temporal photometric consistency

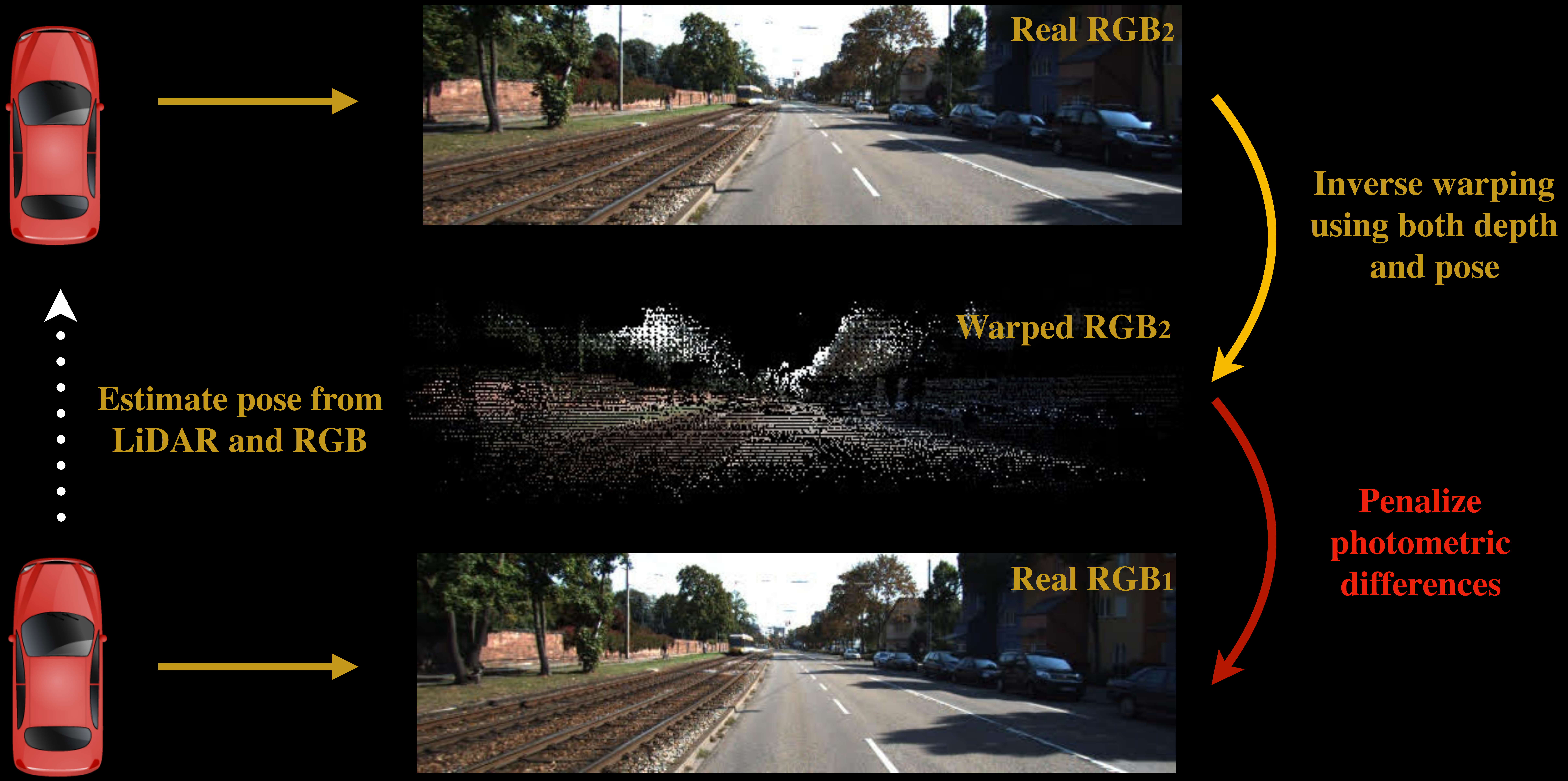
Self-supervision: enforce temporal photometric consistency



Self-supervision: enforce temporal photometric consistency



Self-supervision: enforce temporal photometric consistency

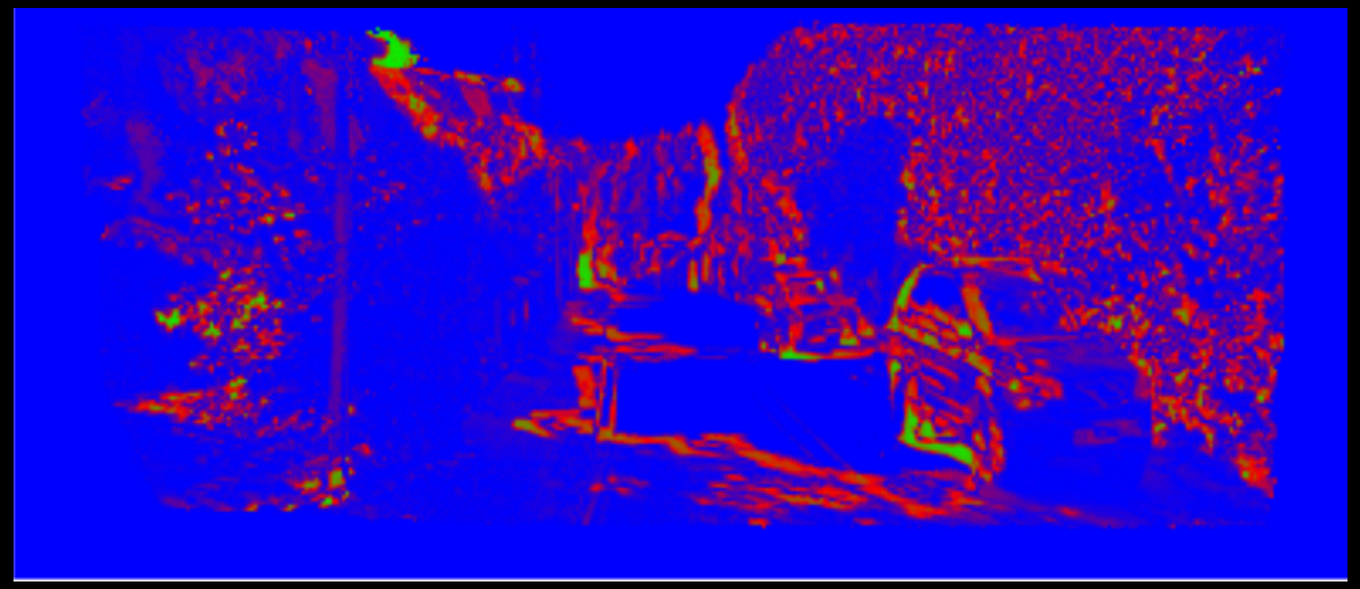


Self-supervision: temporal photometric consistency

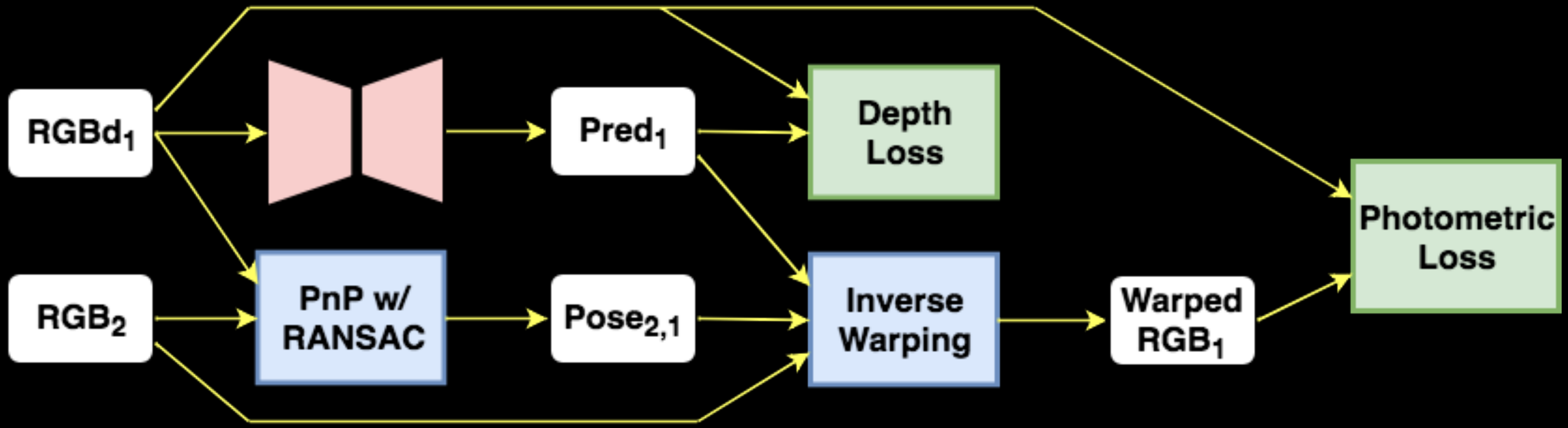
Supervised training requires ground truth depth labels, which are hard to acquire in practice

RGB1

Warped RGB1



Photometric error



Experiment 2. Self-Supervised. RMSE=1.30m

Self-supervised Sparse-to-Dense:
Self-supervised Depth Completion from LiDAR and Monocular Camera

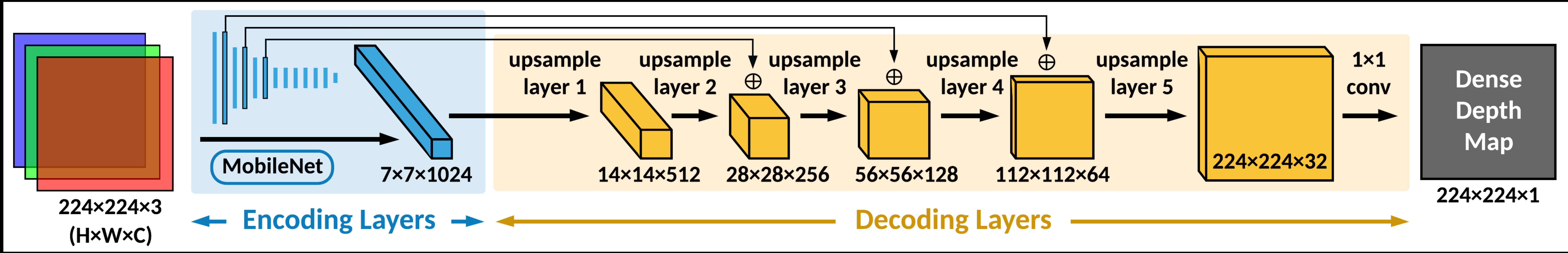
Fangchang Ma, Guilherme Venturelli Cavalheiro, Sertac Karaman

ICRA 2019
code: github.com/fangchangma/self-supervised-depth-completion

Single-View Depth Image Estimation

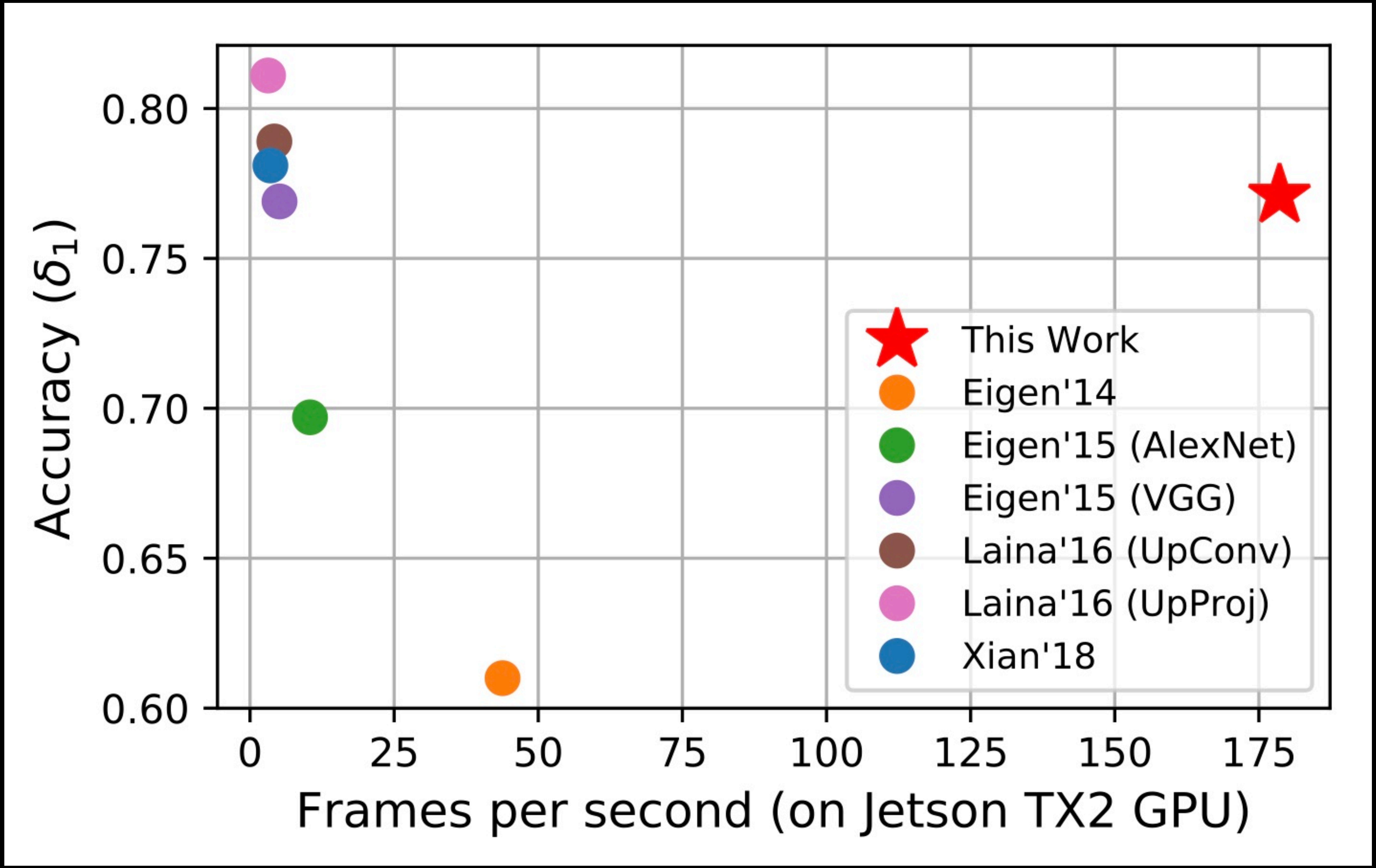
- Why is the problem challenging?
- How to solve the problem?
- How to train a model without ground truth?
- How fast can we run on embedded systems?
- How to obtain performance guarantees with DL?
- What to do if you “hate” deep learning?

FastDepth

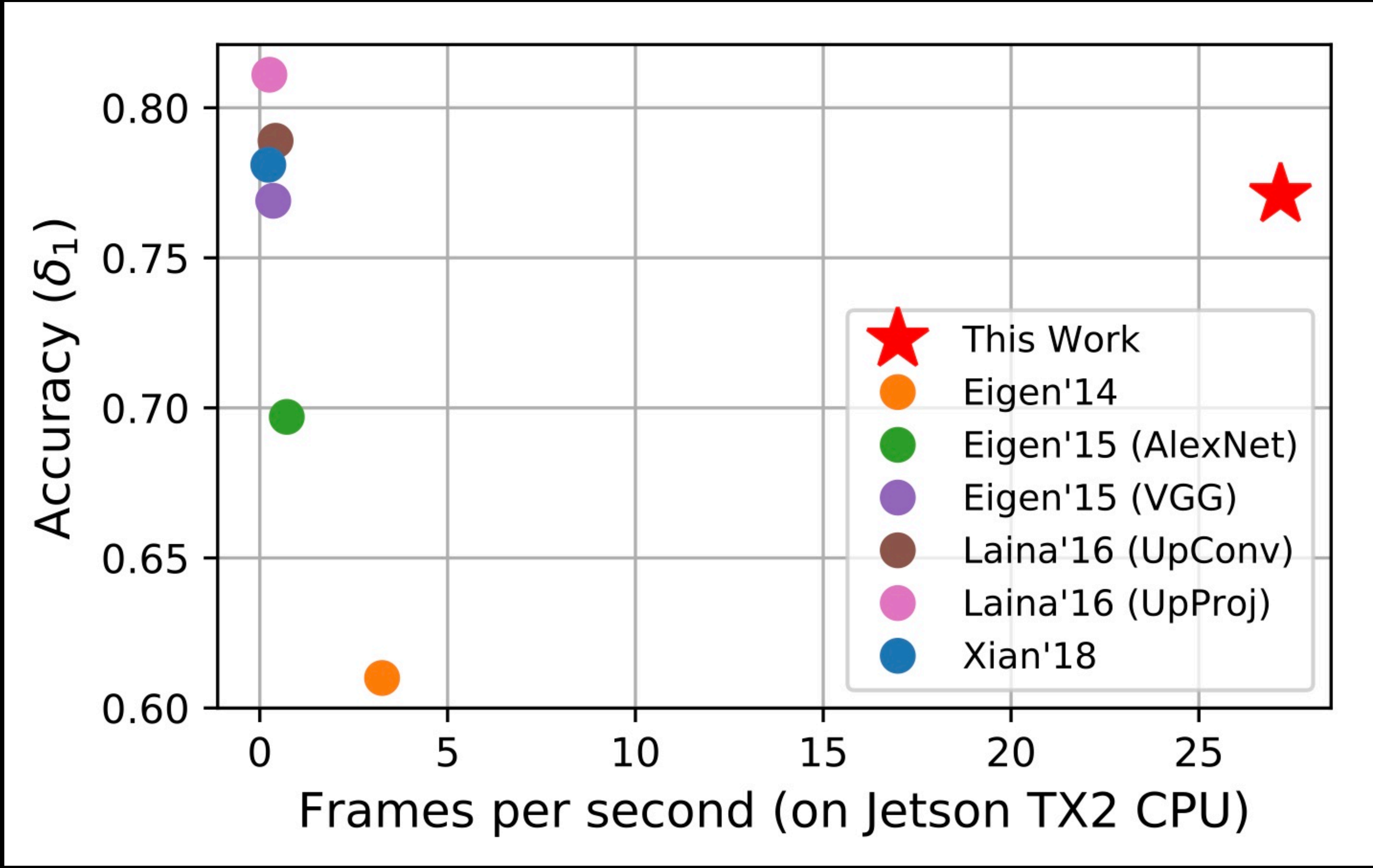


- An Efficient and lightweight encoder-decoder network architecture with a low-latency design incorporating depthwise separable layers and additive skip connections
- Network pruning applied to whole encoder-decoder network
- Platform-specific compilation targeting embedded systems

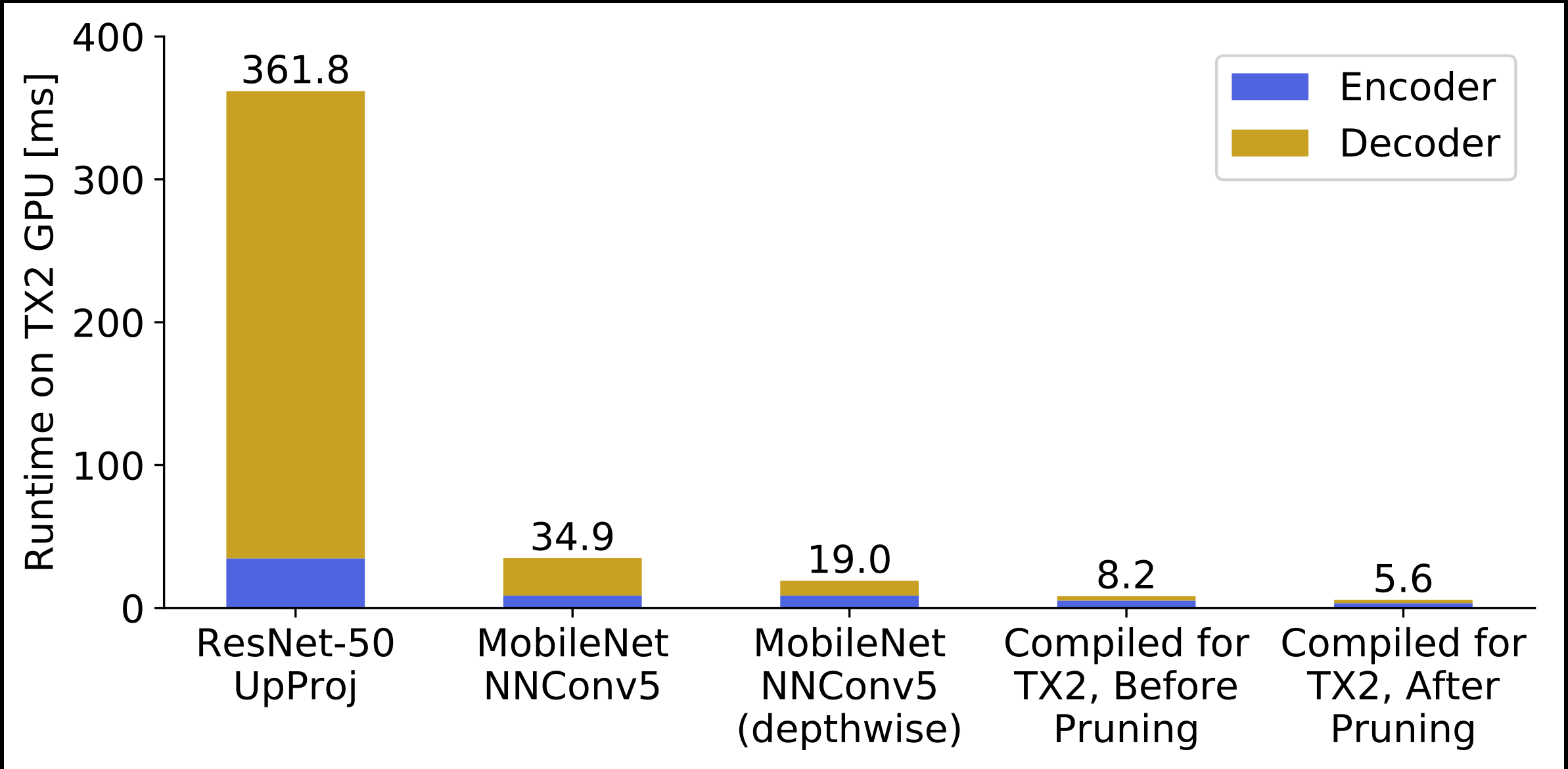
FastDepth is the first demonstration of real-time depth estimation on embedded systems



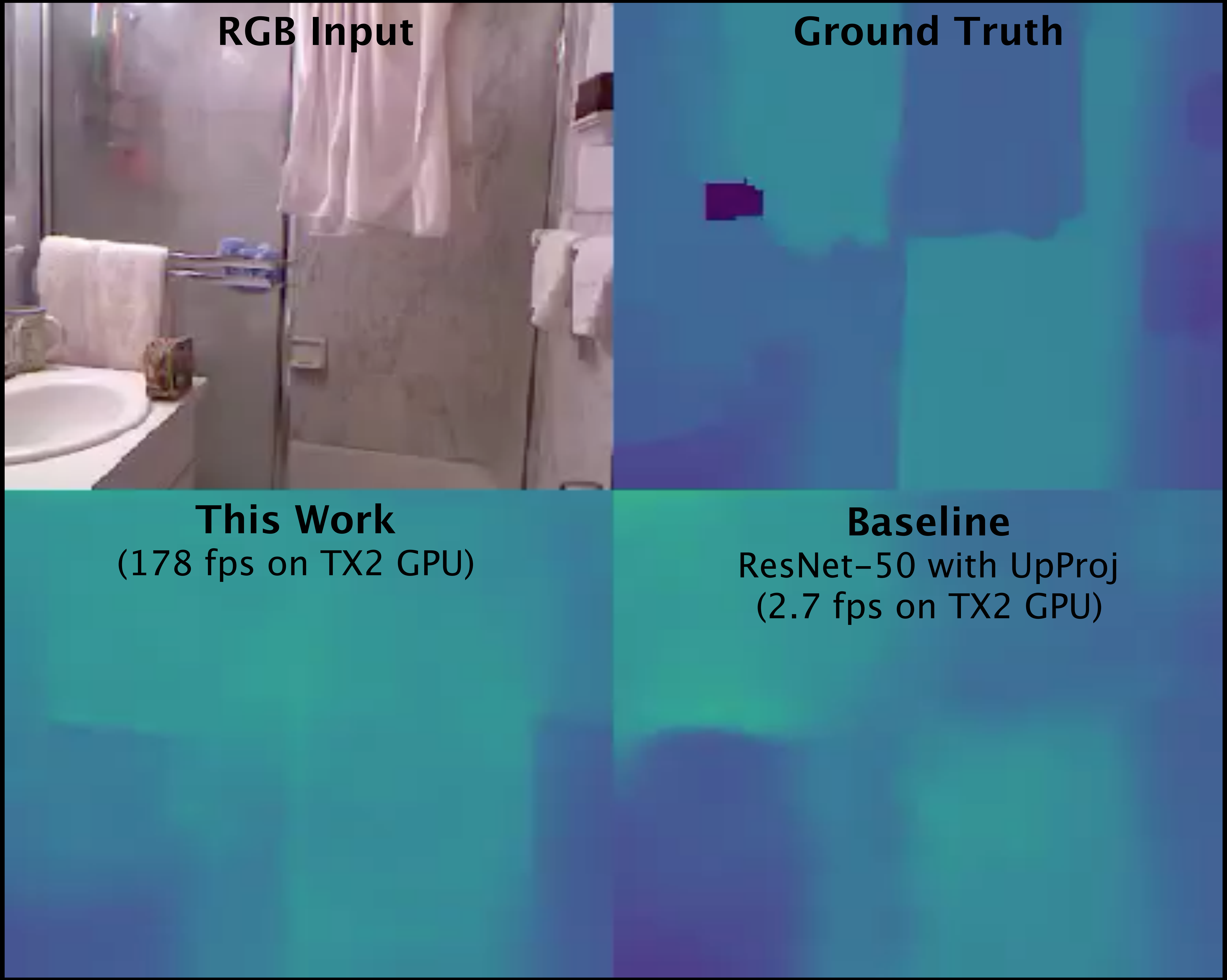
FastDepth is the first demonstration of real-time depth estimation on embedded systems



Achieved fast runtime through network design, pruning, and hardware-specific compilation



FastDepth performs similarly to more complex models, but 65x faster



FastDepth:
Fast Monocular Depth Estimation on Embedded Systems

Diana Wofk*, **Fangchang Ma***, Tienju-Yang, Sertac Karaman, Vivienne Sze

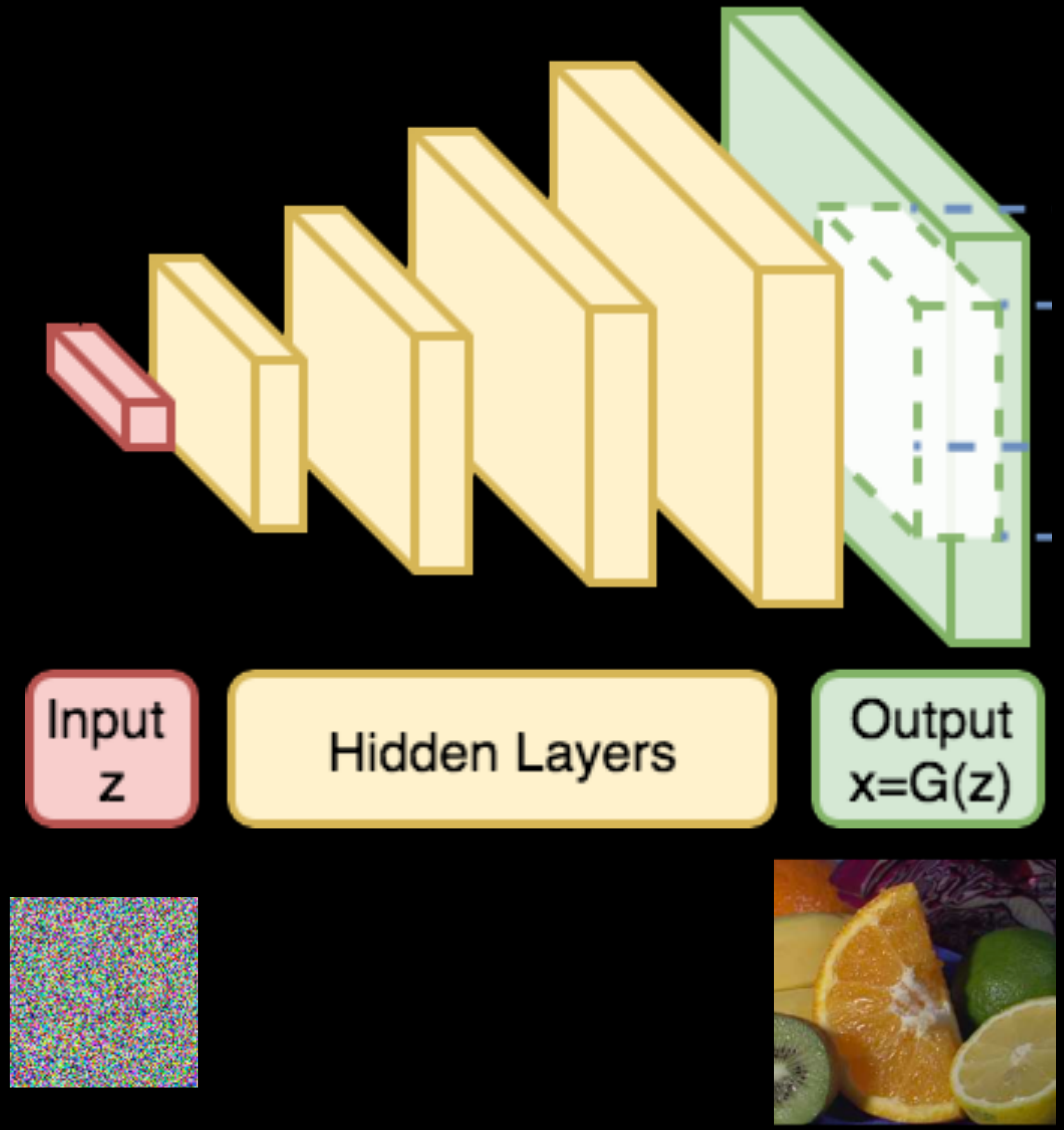
ICRA 2019

fastdepth.mit.edu
https://github.com/dwofk/fast-depth

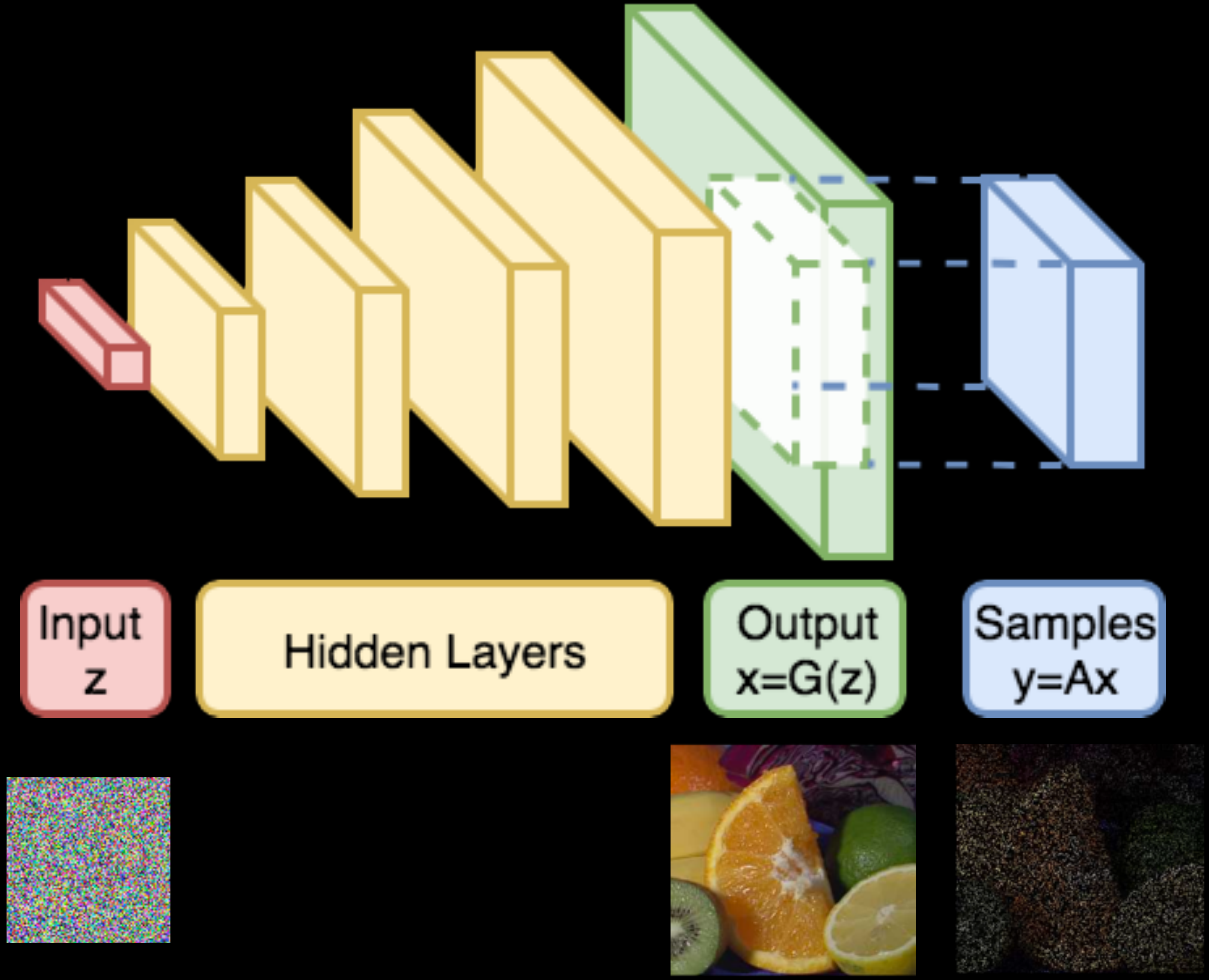
Single-View Depth Image Estimation

- Why is the problem challenging?
- How to solve the problem?
- How to train a model without ground truth?
- How fast can we run on embedded systems?
- How to obtain performance guarantees with DL?
- What to do if you “hate” deep learning?

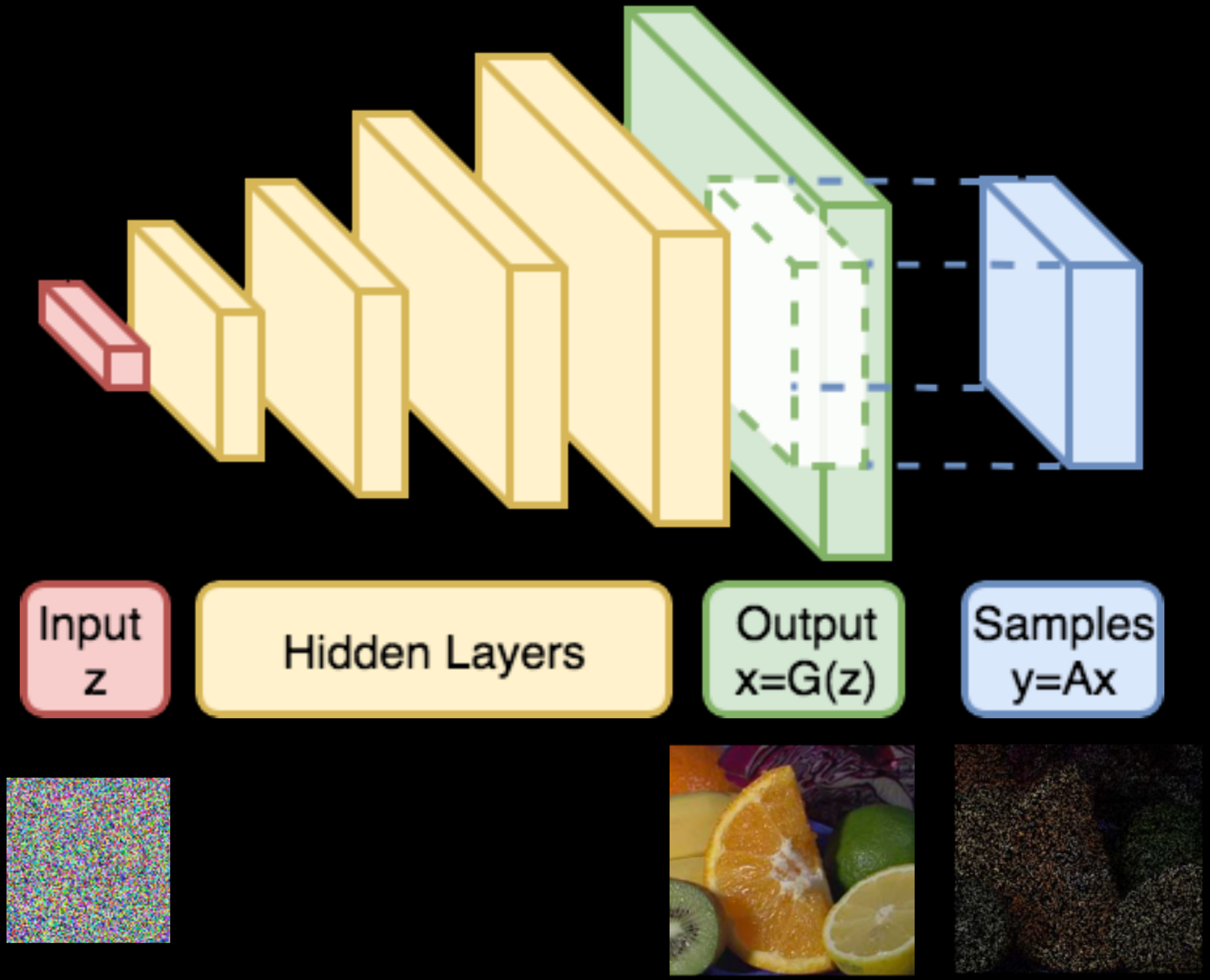
Assumption: image can be modeled by a convolutional generative neural network



Sub-sampling Process

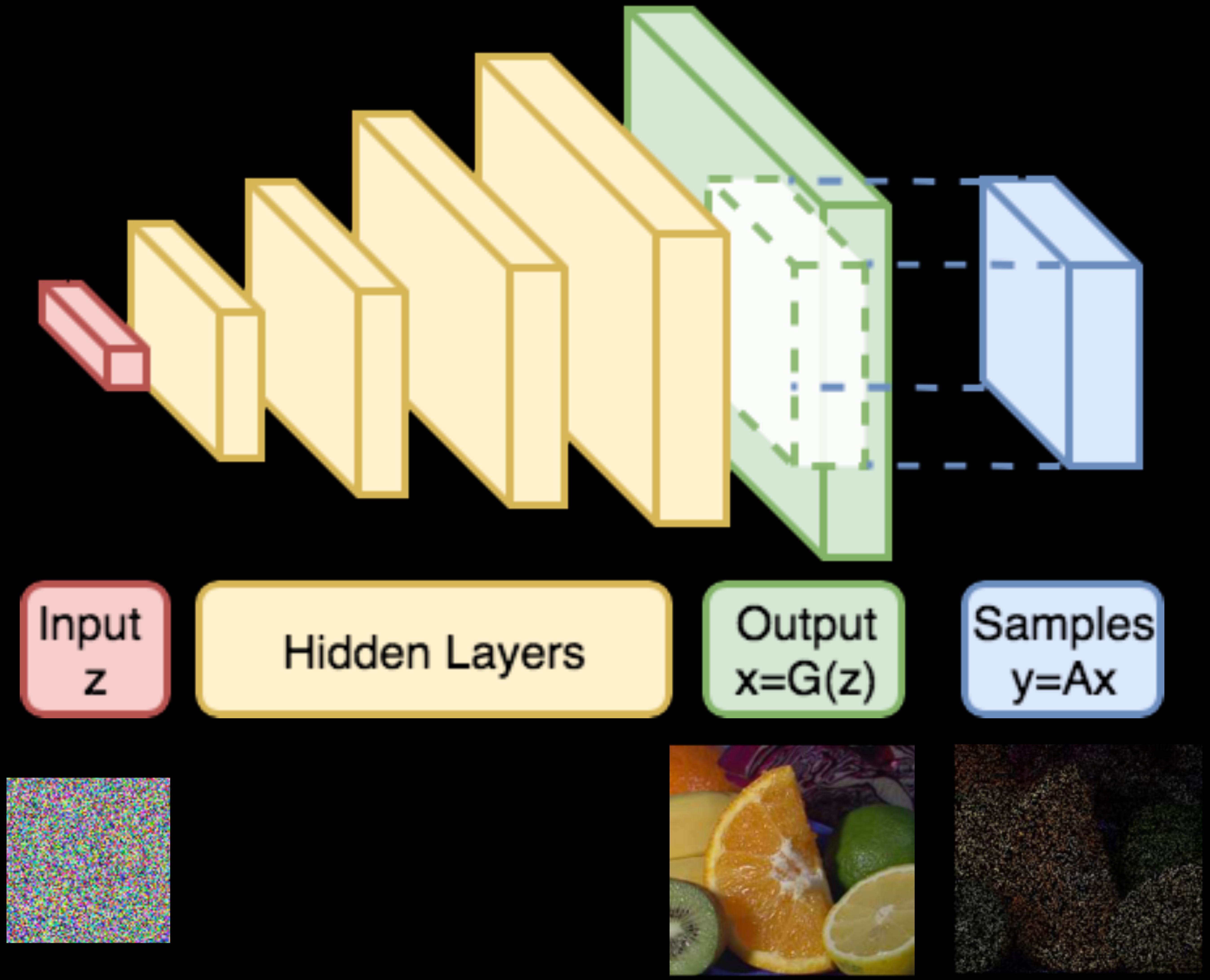


Rephrasing the depth-completion/image-inpainting problems



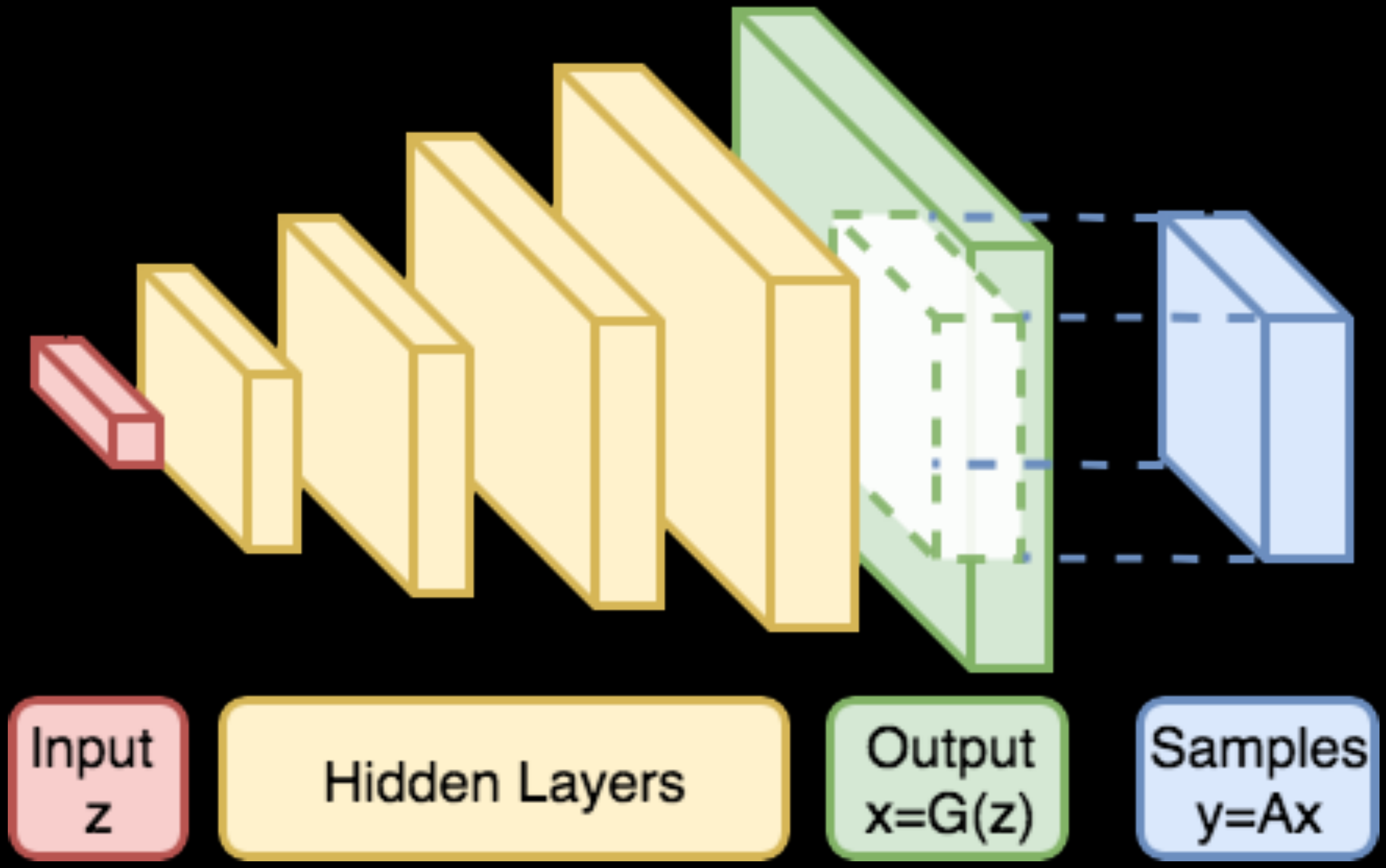
Question: can you find x (or equivalently, z), given only y?

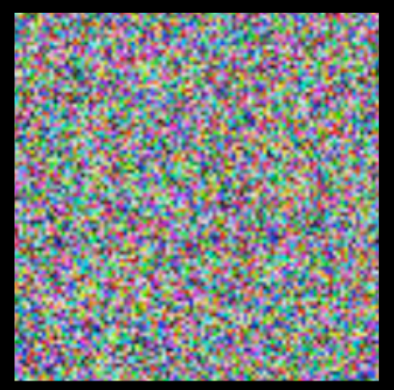
Rephrasing the depth-completion/image-inpainting problems



If z is recovered, then we can reconstruct x as G(z) using a single forward pass

The latent code z can be computed efficiently using gradient descent

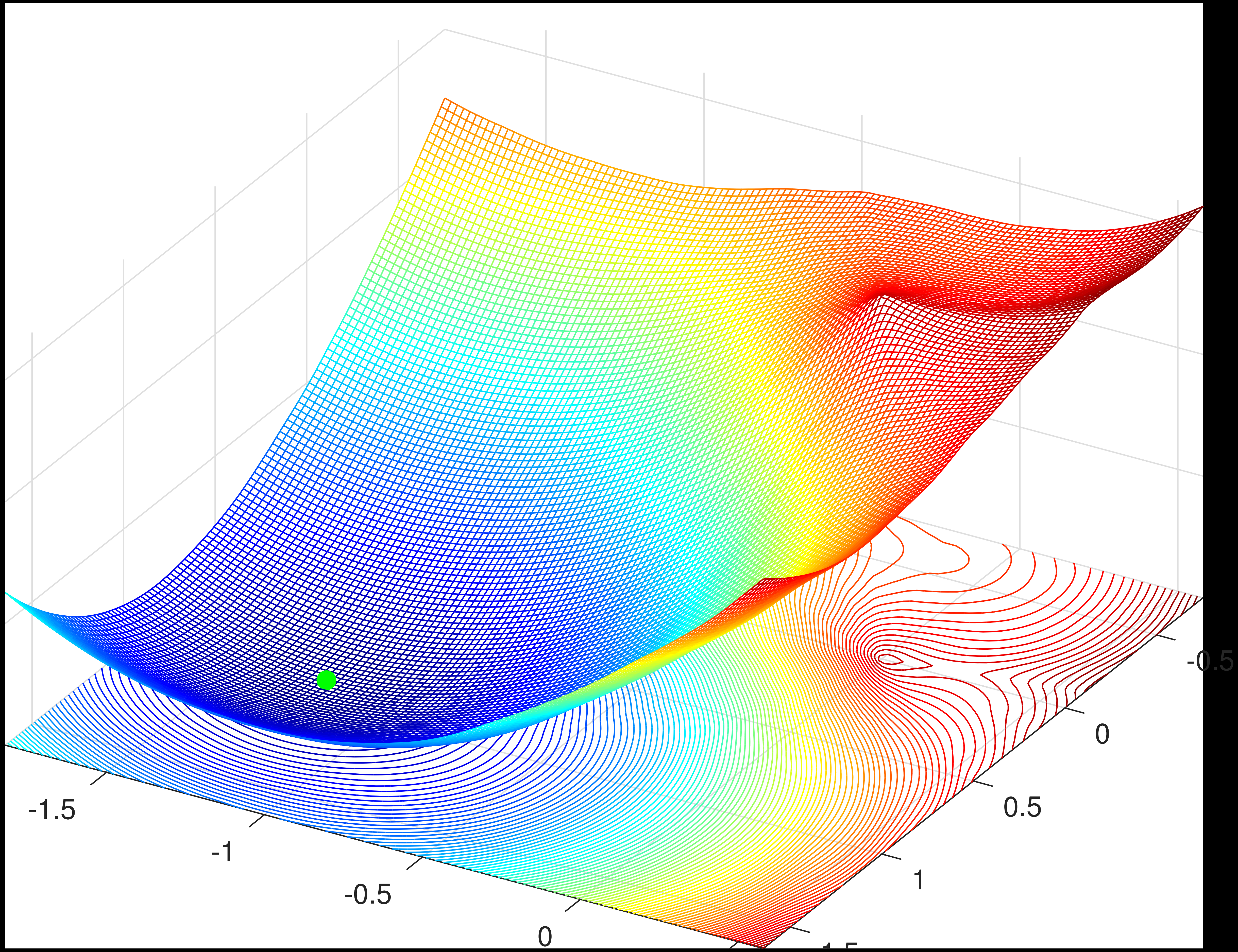




$$\hat{z} = \arg\min_{z} \|G(z) - y\|^2$$

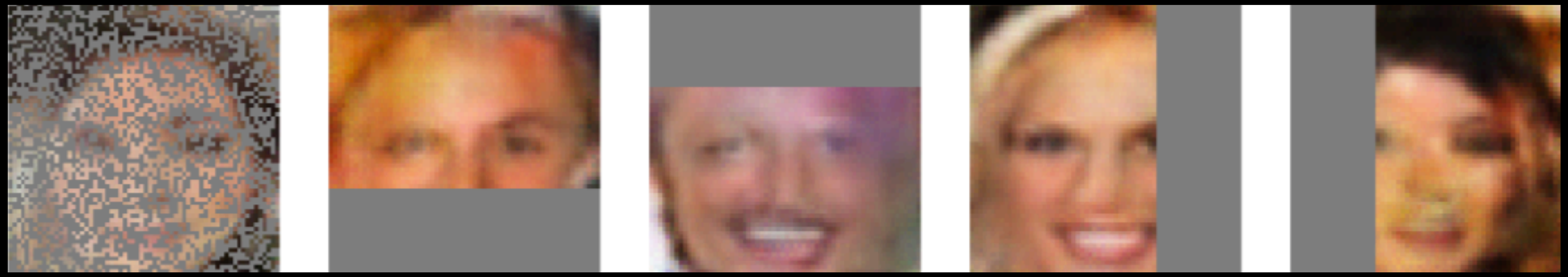
Main Theorem

For a 2-layer network, the latent code z can be recovered from the undersampled measurements y using gradient descents (with high probability) by minimizing the empirical loss function.

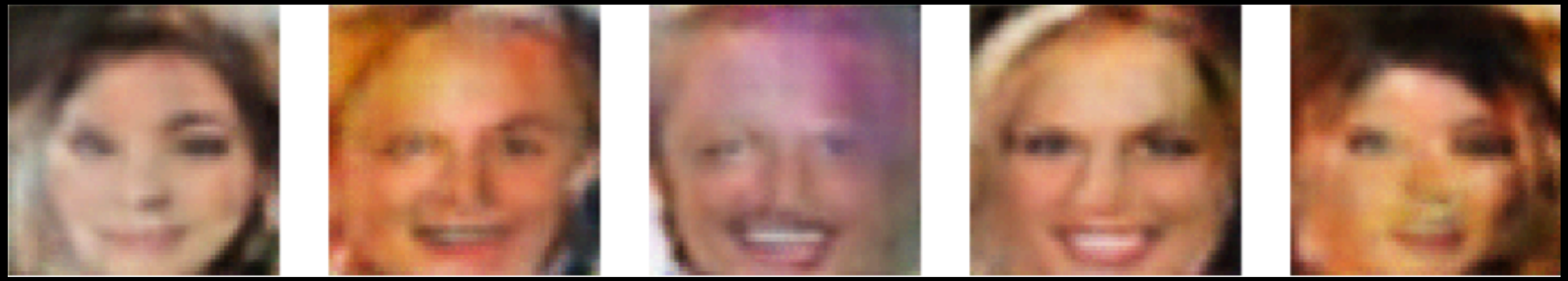


Experimental Results

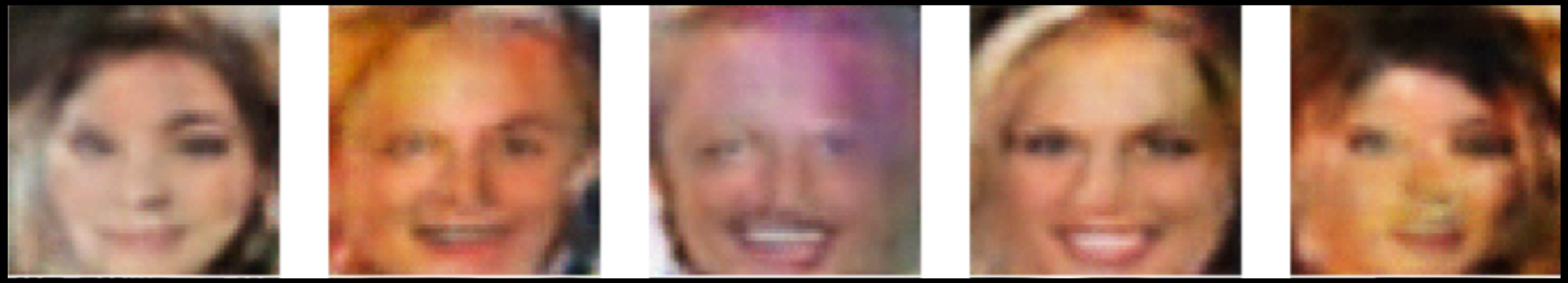
Undersampled Measurements



Reconstructed Images



Ground Truth



Invertibility of Convolutional Generative Networks from Partial Measurements

Fangchang Ma*, Ulas Ayaz*, Sertac Karaman

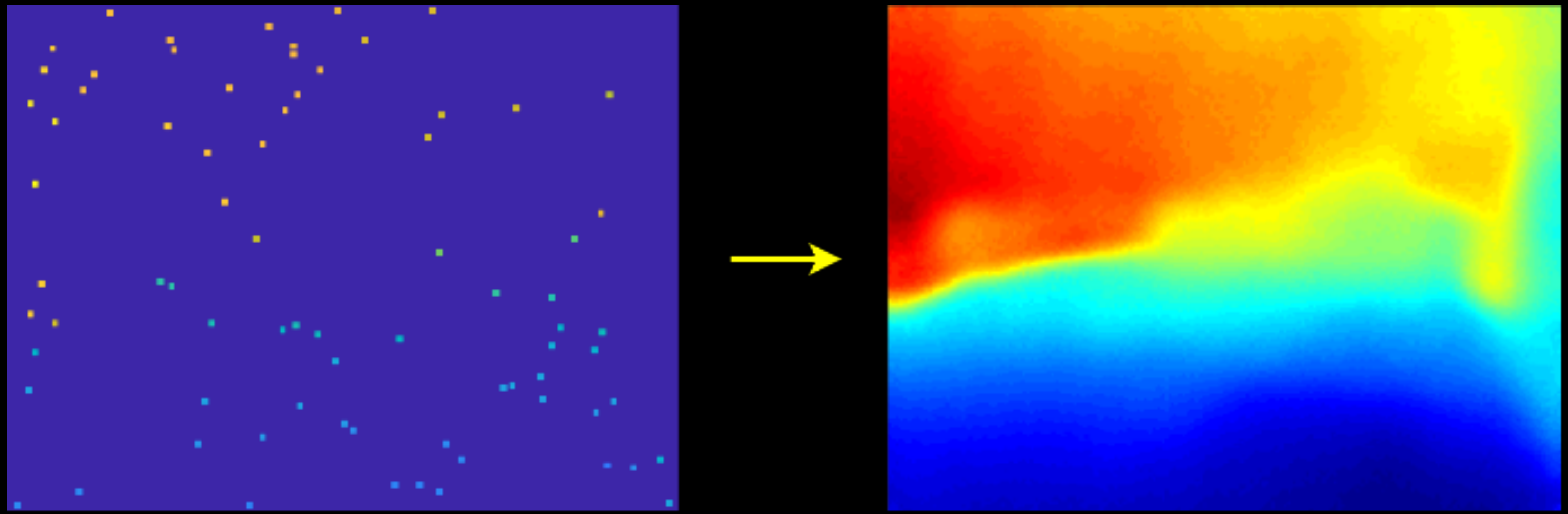
NeurIPS 2018 (previously known as NIPS)
code: github.com/fangchangma/invert-generative-networks

MIT AEROASTRO LIDS

Single-View Depth Image Estimation

- Why is the problem challenging?
- How to solve the problem?
- How to train a model without ground truth?
- How fast can we run on embedded systems?
- How to obtain performance guarantees with DL?
- What to do if you “hate” deep learning?

Depth Completion:
Linear model with planar assumption



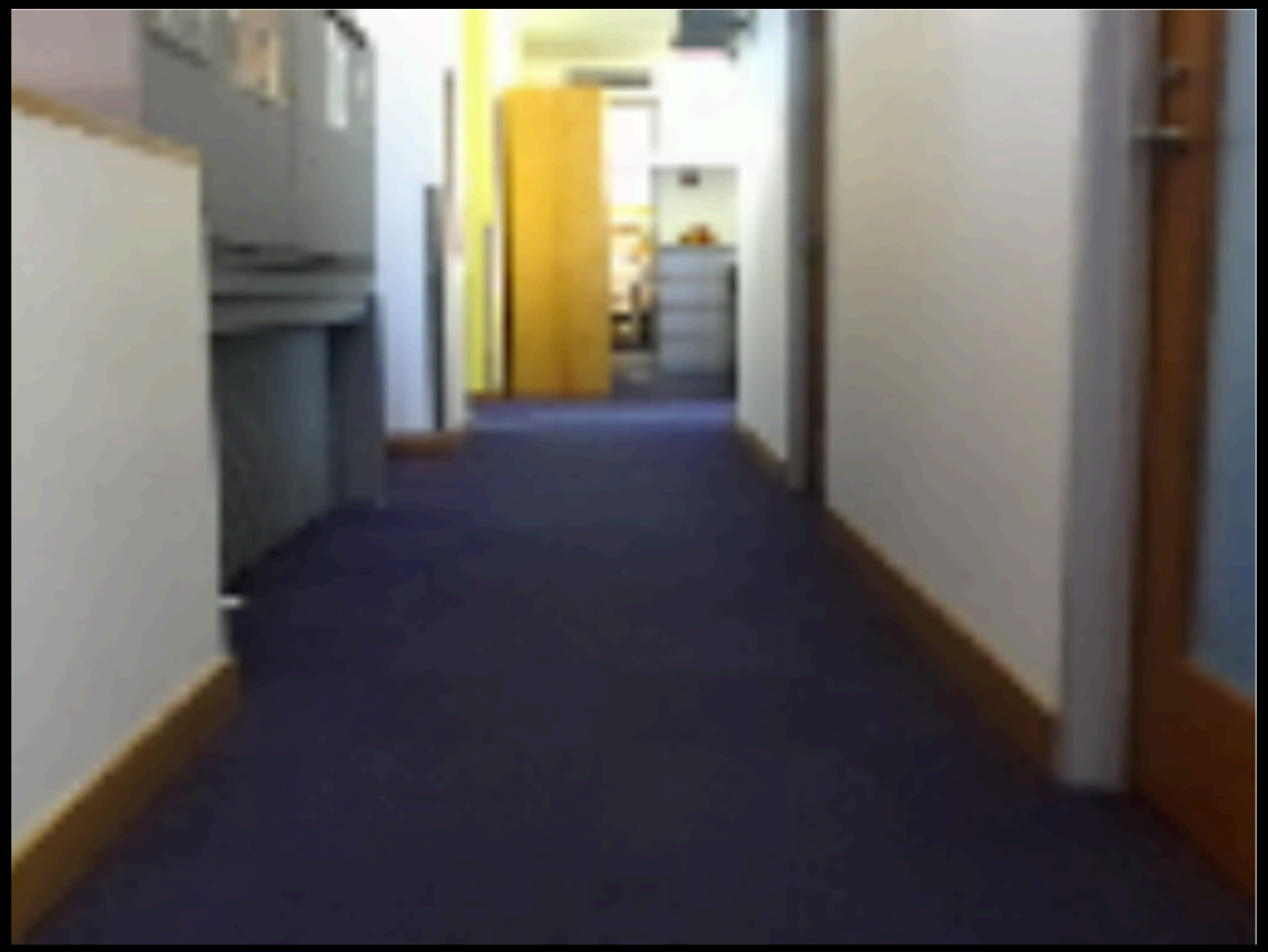
Input: only sparse depth
Output: dense depth

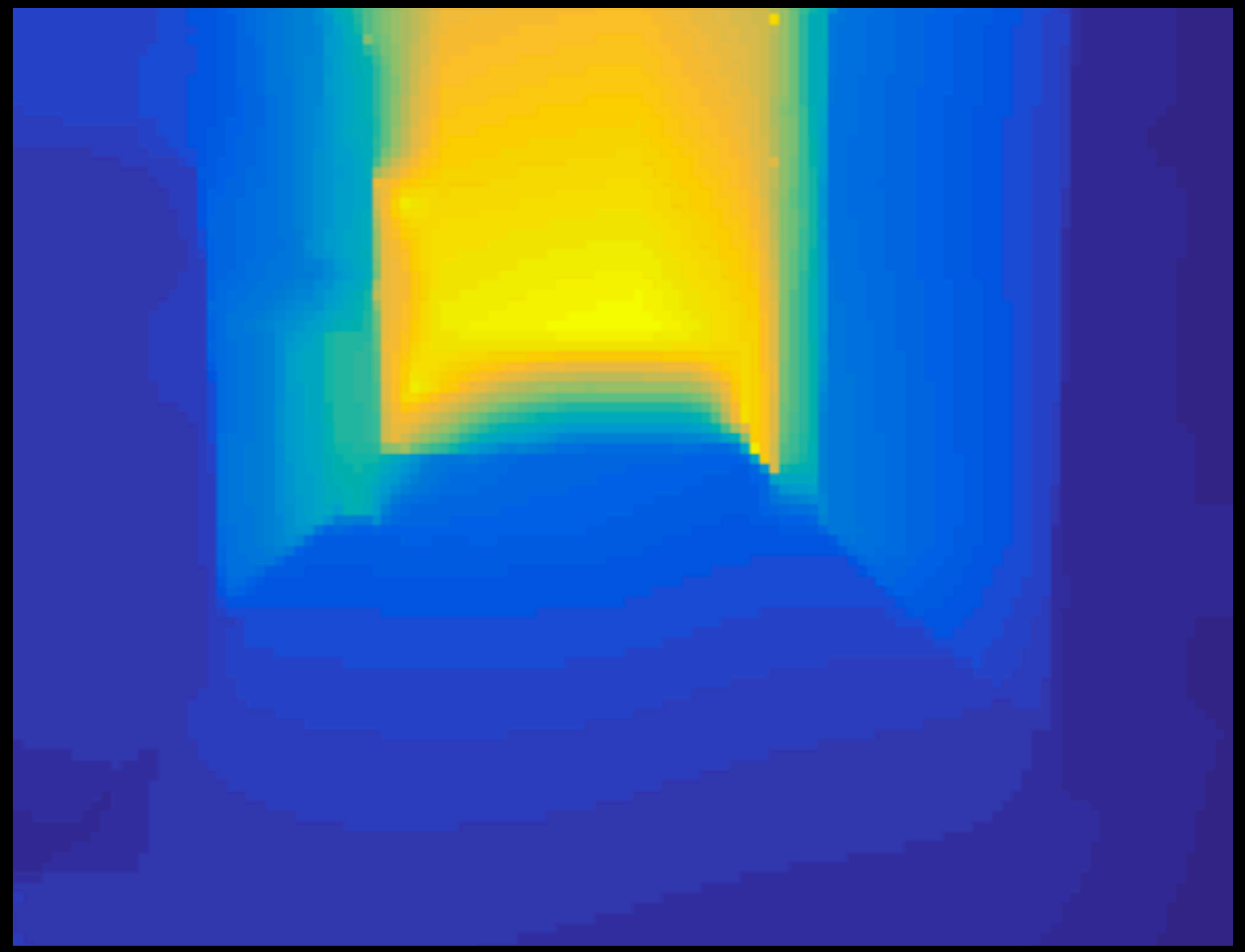
Fangchang Ma, Luca Carlone, Ulas Ayaz, Sertac Karaman. “Sparse sensing for resource-constrained depth reconstruction”. *IROS’16*

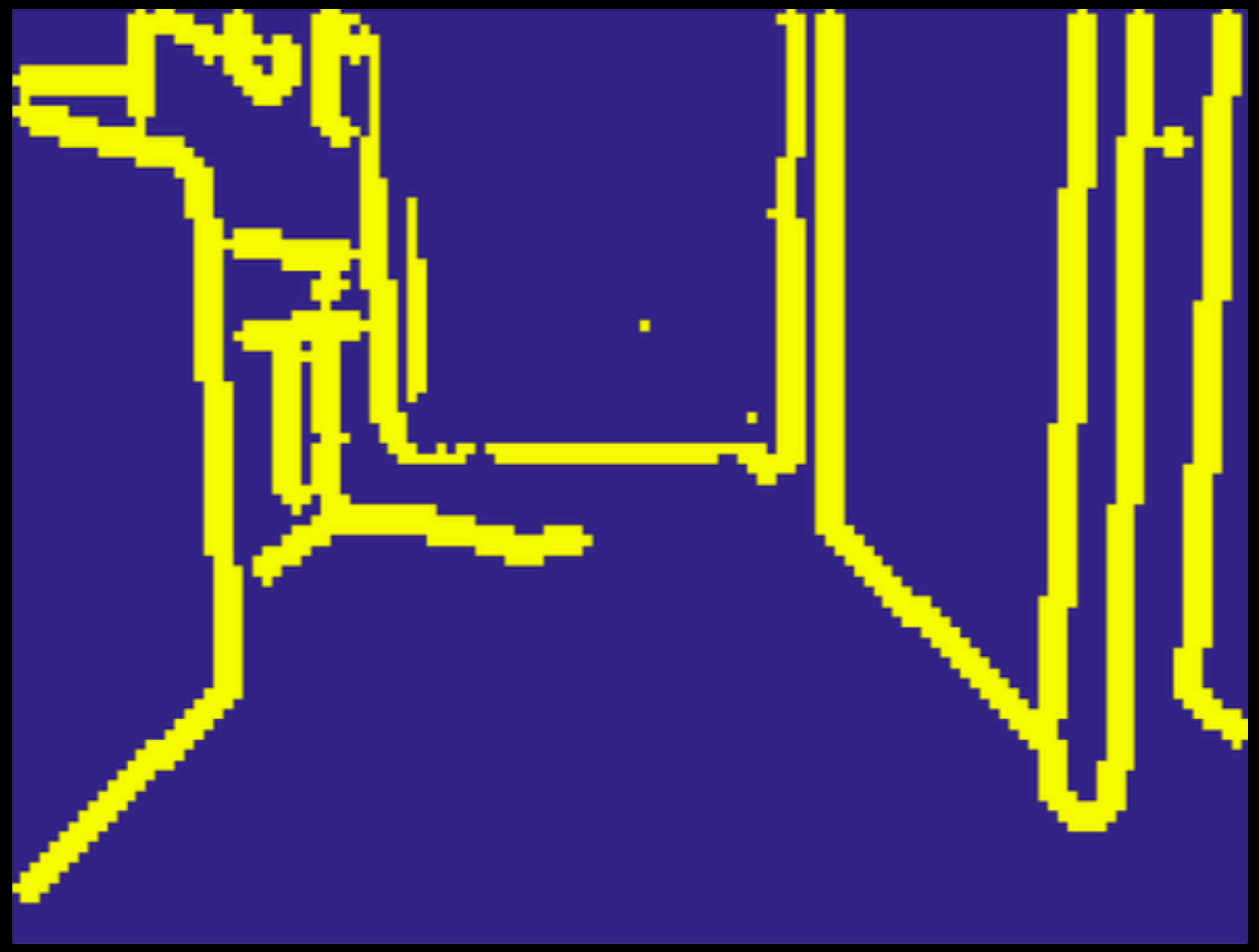
Fangchang Ma, Luca Carlone, Ulas Ayaz, Sertac Karaman. “Sparse Depth Sensing for Resource-Constrained Robots”. *The International Journal of Robotics Research (IJRR)*

Depth Completion: Linear model with planar assumption

Planar Assumption: a relatively structured environment
can be well approximated by a small number of planar surfaces







Observation: 2nd derivative of planar surfaces is sparse

Implication: 2nd derivative of a structured environment is approximately sparse
(sparsity of 2nd derivative is a measure of scene complexity)

Depth Completion: Linear model with planar assumption

Planar Assumption: sparse 2nd derivative in a typical depth image

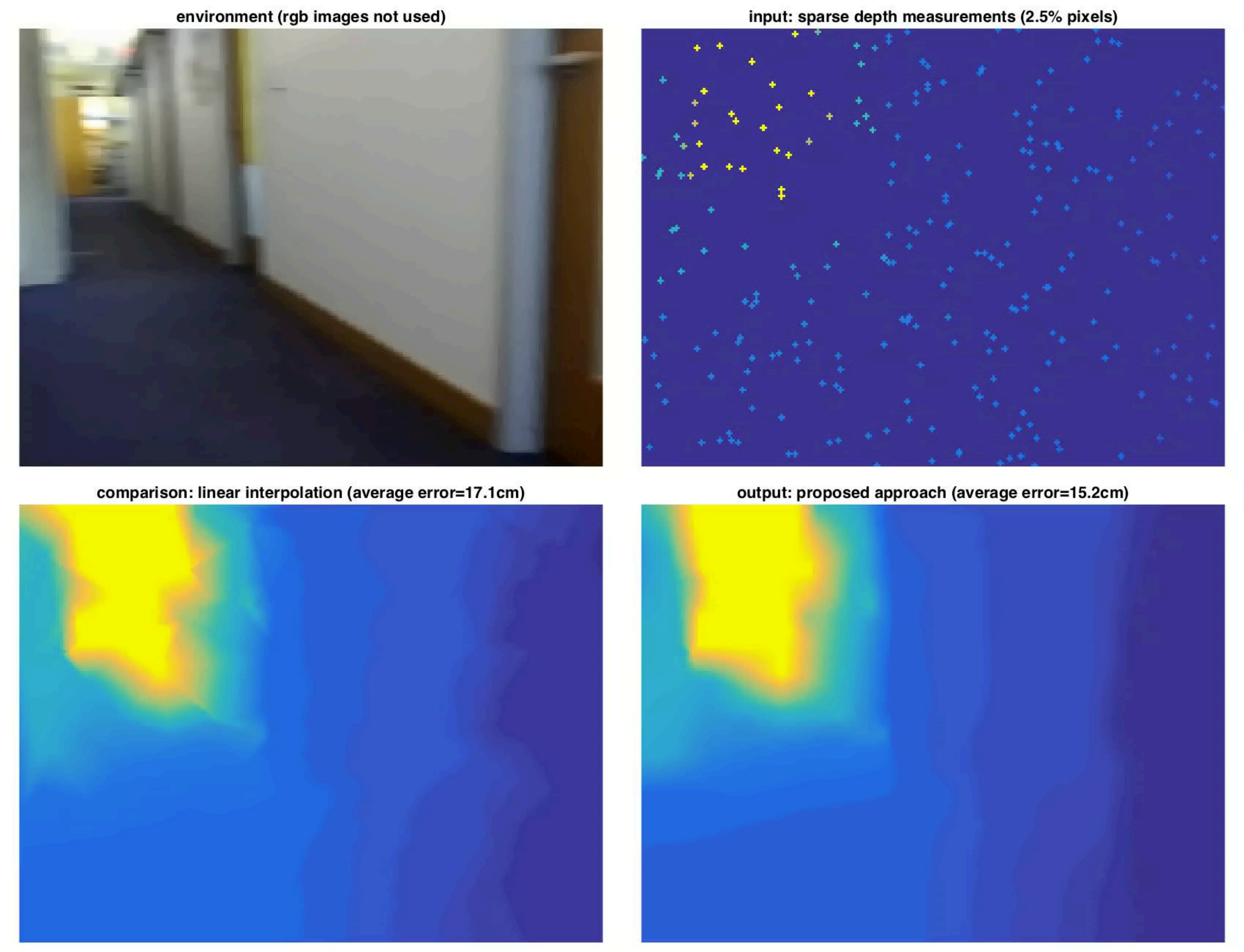
Goal: find the simplest depth image (with the sparsest 2nd derivative) that is aligned with our measurements

$$\min_x \|\Delta x\|_0, \quad \text{subject to } y = Ax$$

Convex Relaxation (Linear Programming):

$$\min_x \|\Delta x\|_1, \quad \text{subject to } y = Ax$$

Depth Completion: Linear model with planar assumption



Sparse Depth Sensing for Resource-Constrained Robots

Fangchang Ma*, Ulas Ayaz*, Sertac Karaman

The International Journal of Robotics Research (IJRR)
code: github.com/sparse-depth-sensing/sparse-depth-sensing

Single-View Depth Image Estimation

Fangchang Ma

homepage: www.mit.edu/~fcma/
code: github.com/fangchangma

- Why is the problem challenging?
- How to solve the problem?
- How to train a model without ground truth?
- How fast can we run on embedded systems?
- How to obtain performance guarantees with DL?
- What to do if you “hate” deep learning?