CS231A

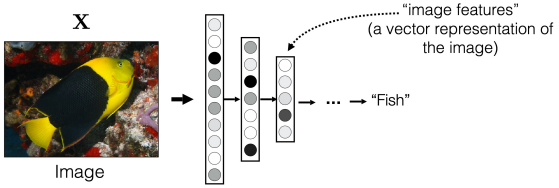
Computer Vision:

From 3D Reconstruction to Recognition

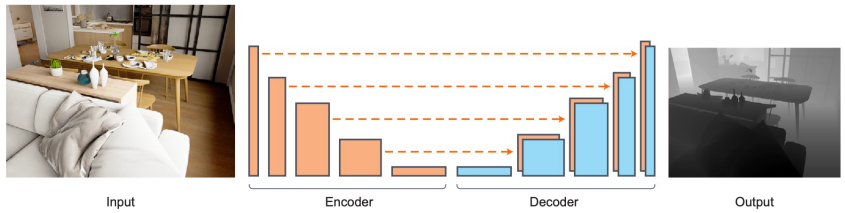
Monocular Depth Estimation & Feature Tracking

Learning Goals for Upcoming Lectures

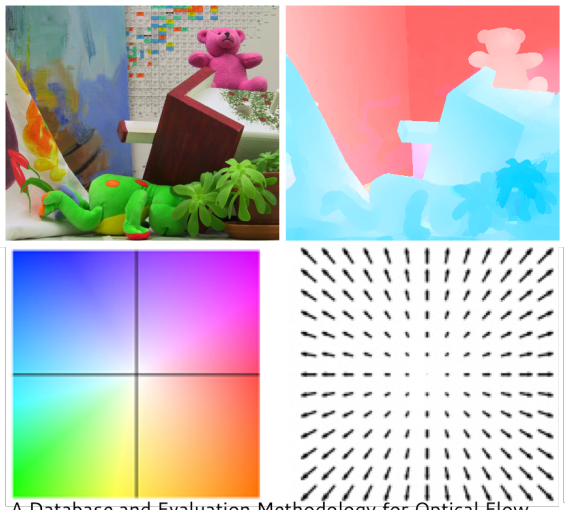
Representations & Representation Learning



Monocular Depth Estimation, Feature Tracking

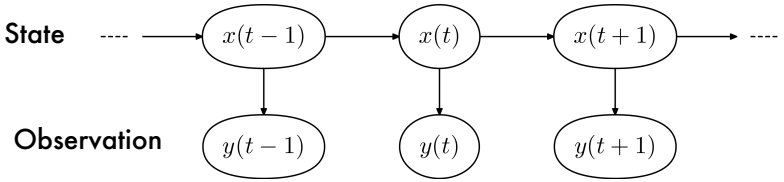


Optical & Scene Flow

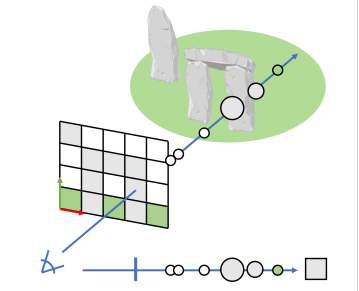


A Database and Evaluation Methodology for Optical Flow.
Baker et al. IJCV. 2011

Optimal Estimation



Neural Radiance Fields



Outline of the previous lecture

- What is a state? What is a representation?
- What are the different kinds of representations?
- How can we extract state from raw sensory data?
- What kind of data can we process?

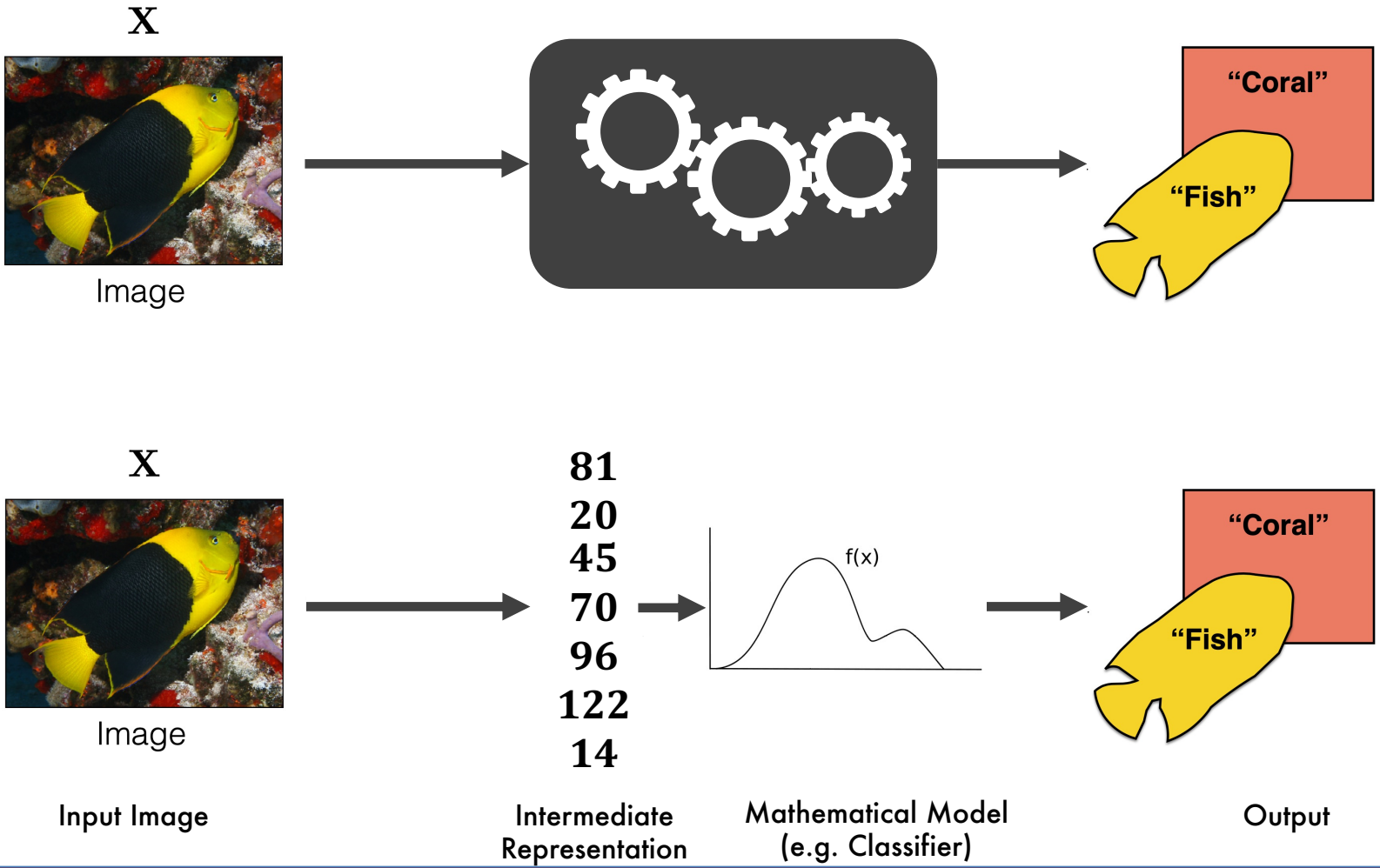
Summary of what you learned

- **State:** Quantity that describes the most important aspect of a dynamical system at time t
- **Representation:** data format of input or output including a low-dimensional representation of sensor data
  - Input/output/intermediate representation

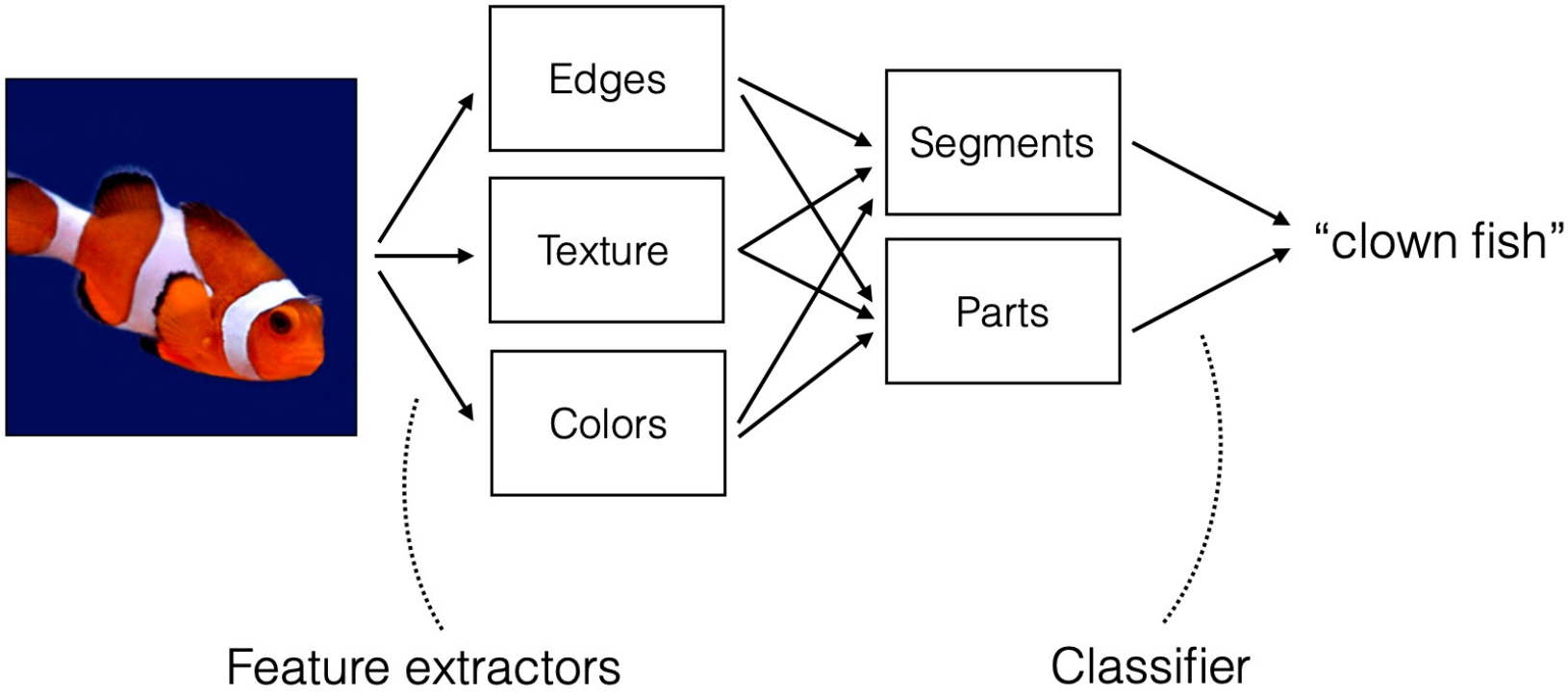
Summary of what you learned

- Learned versus interpretable representations
- Visualize learned representations
- How to learn representations?
  - Supervised
  - Unsupervised
  - Self-supervised

Typical CV Pipeline

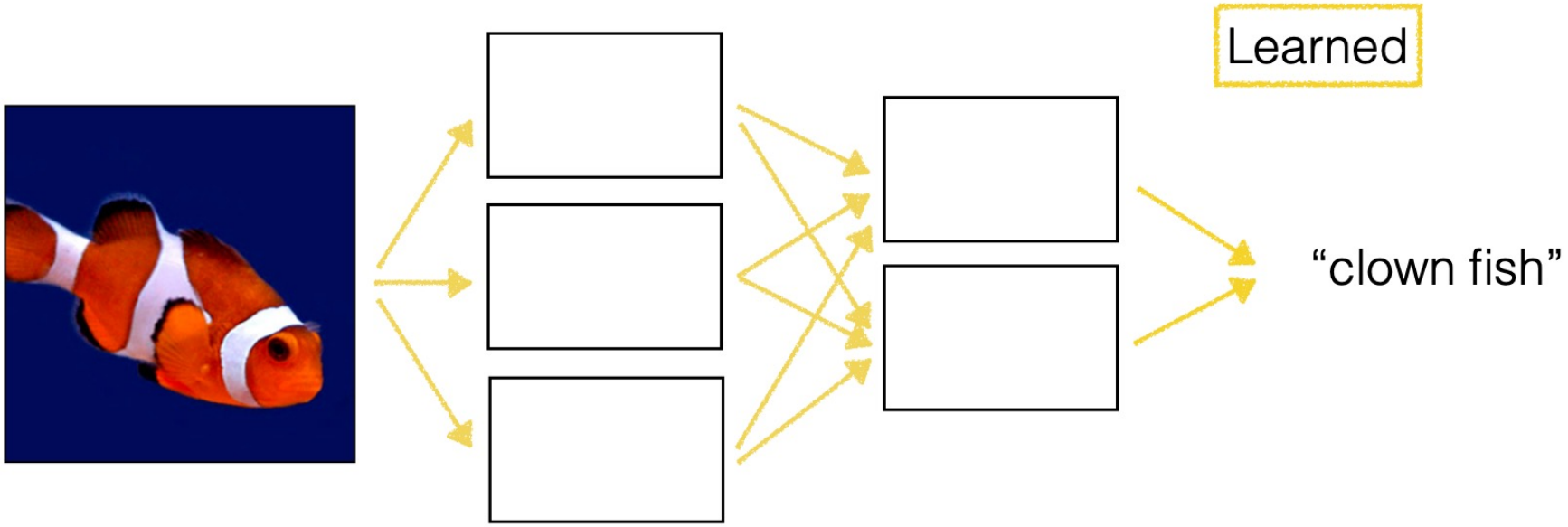


Traditional CV Pipeline



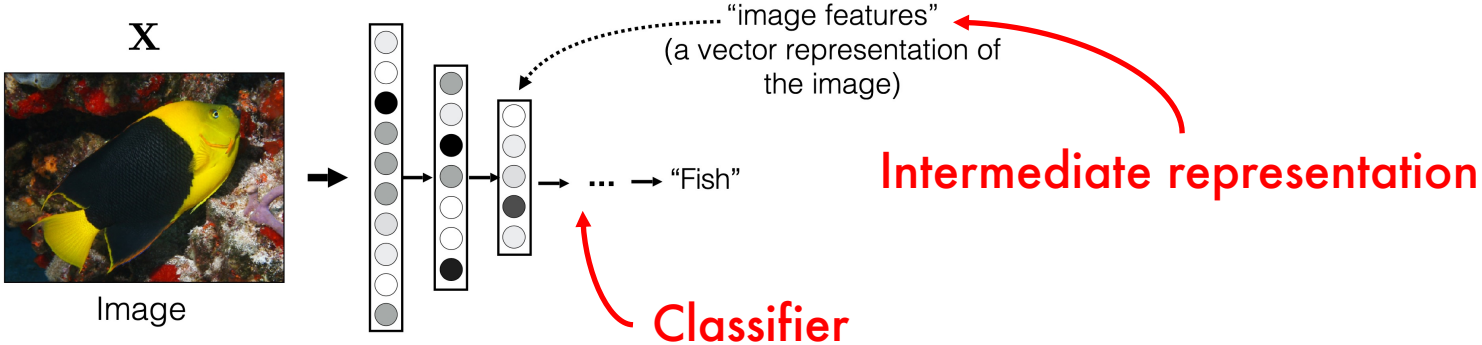
Example from Advances in Computer Vision – MIT – 6.869/6.819

Traditional CV Pipeline



Example from Advances in Computer Vision – MIT – 6.869/6.819

Supervised learning of a representation



Training data

Input images

Labels

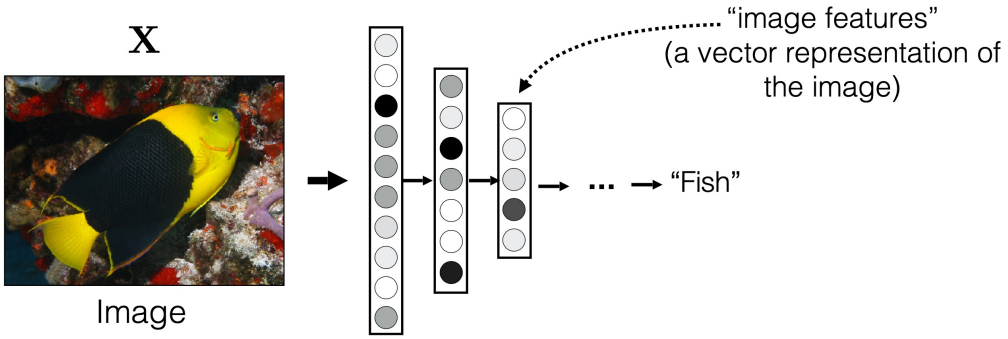
$$\{x_1, y_1\}$$
$$\{x_2, y_2\}$$
$$\{x_3, y_3\}$$
$$\dots$$

$$\rightarrow \quad \text{Learner} \quad \rightarrow \quad f : X \rightarrow Y$$

$$f^* = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{N} \mathcal{L}(f(x_i), y_i)$$

Loss function/Cost

Learning without Labels



Training data

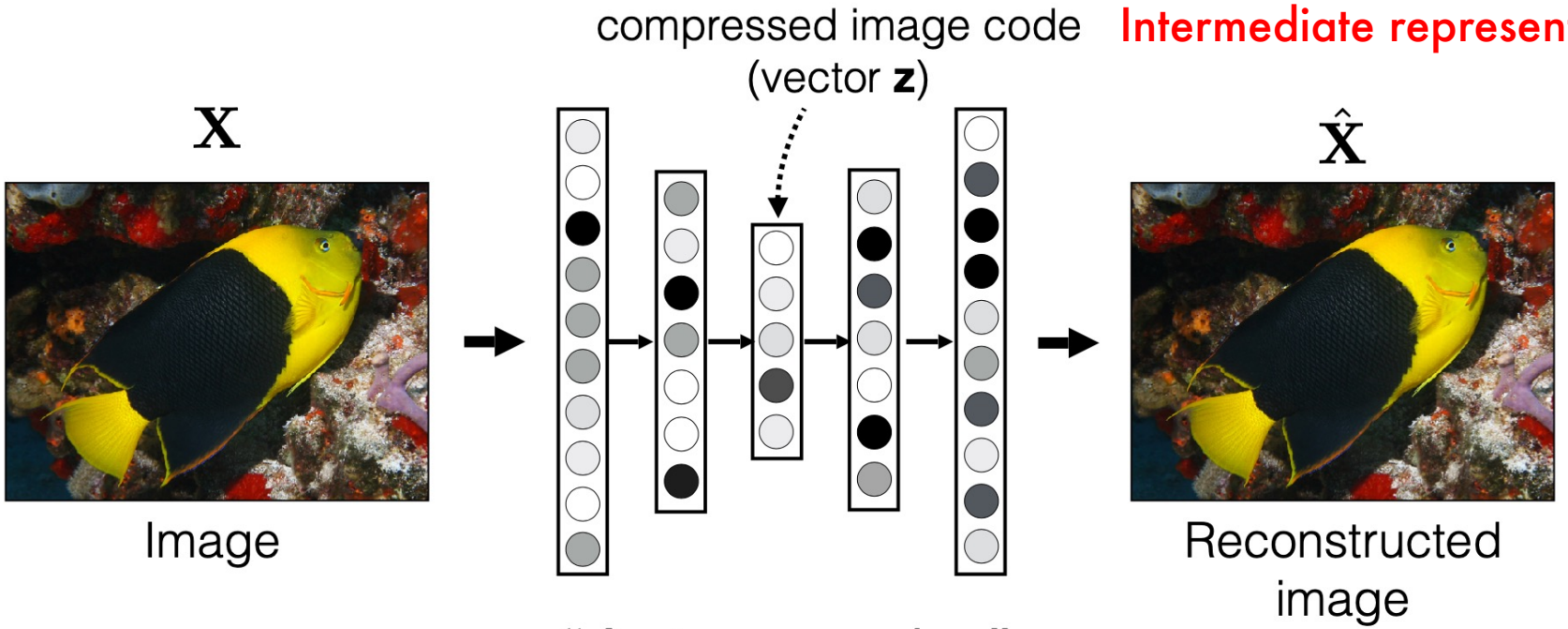
$$\{x_1, y_1\}$$
$$\{x_2, y_2\}$$
$$\{x_3, y_3\}$$
$$\dots$$

$$\rightarrow$$ [crossed out] $$\rightarrow f : X \rightarrow Y$$

$$f^* = \arg \min_{f \in \mathcal{F}} \sum_{i=1} \mathcal{L}(f(\quad, y_i)$$

Unsupervised Representation Learning

No category or symbolic label. Instead: learn to reconstruct.

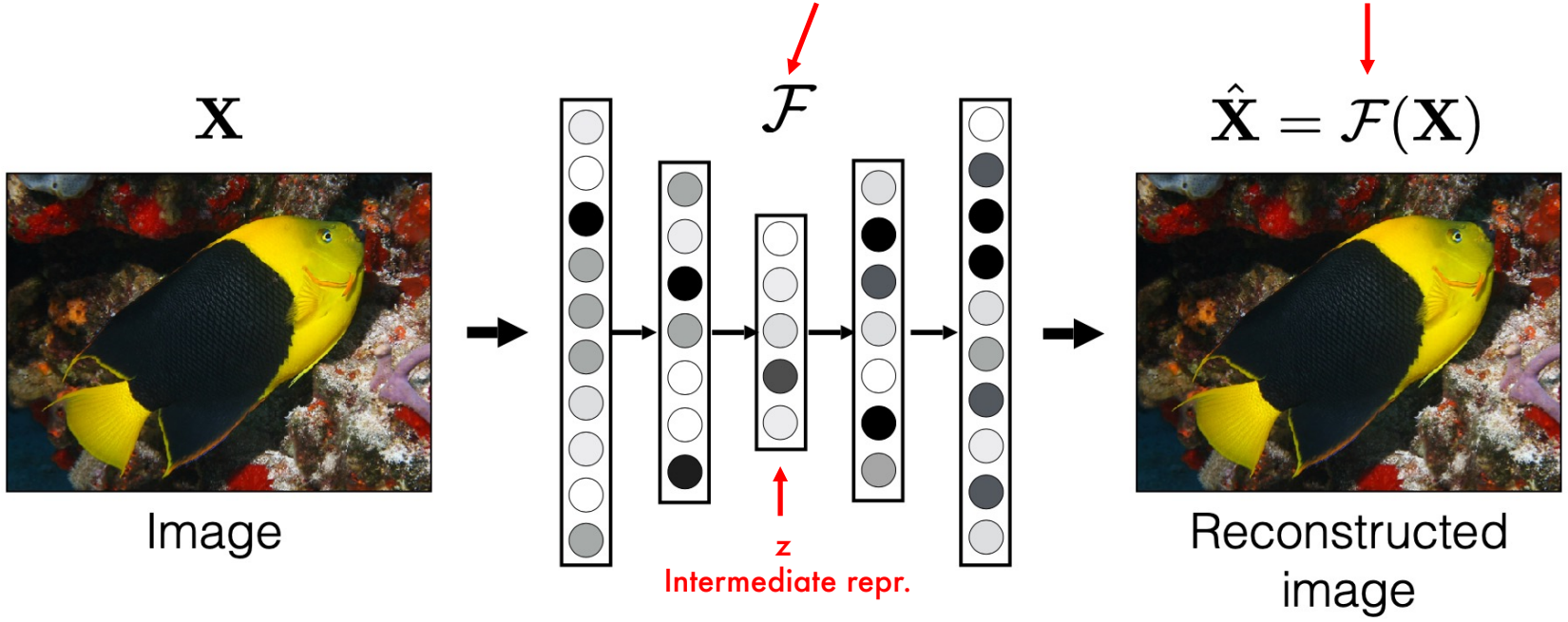


One kind of unsupervised model: "Autoencoder"

[e.g., Hinton & Salakhutdinov, Science 2006]

Autoencoder

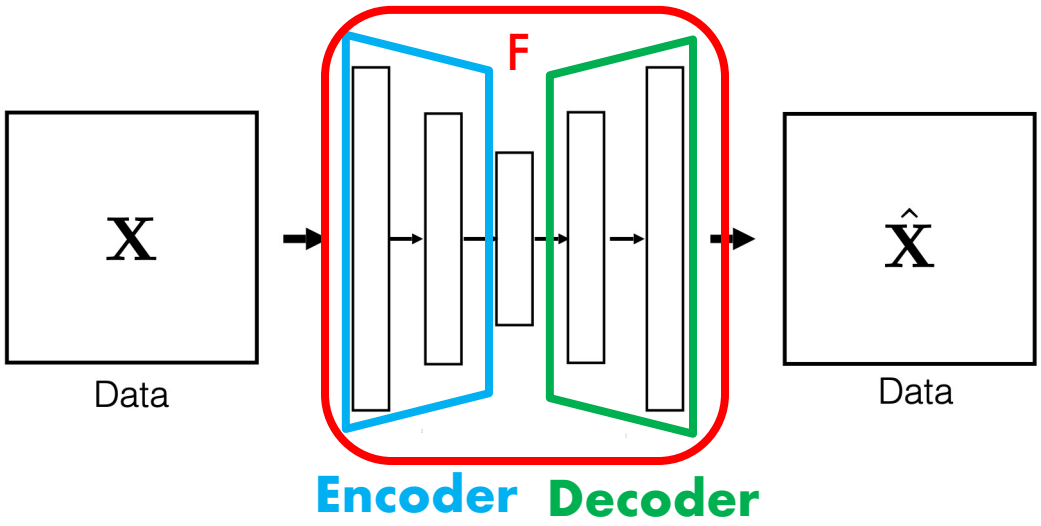
Not the Fundamental Matrix, but a function representing the autoencoder

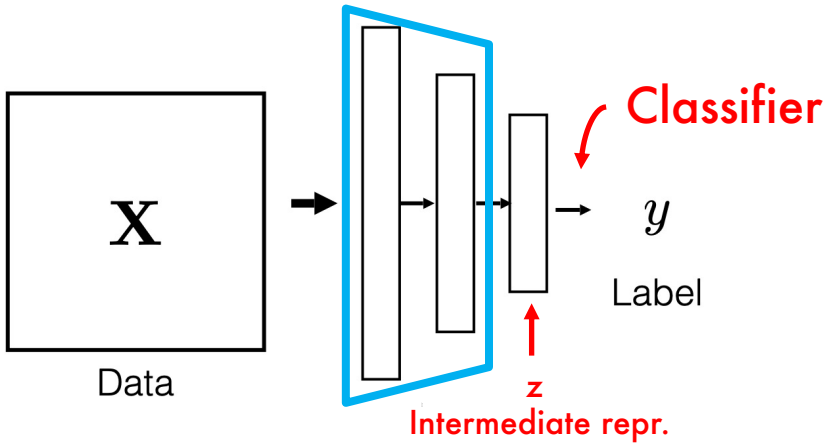


$$\arg\min_{\mathcal{F}} \mathbb{E}_{\mathbf{X}}[||\mathcal{F}(\mathbf{X}) - \mathbf{X}||]$$

Reconstruction loss to minimize by finding optimal F

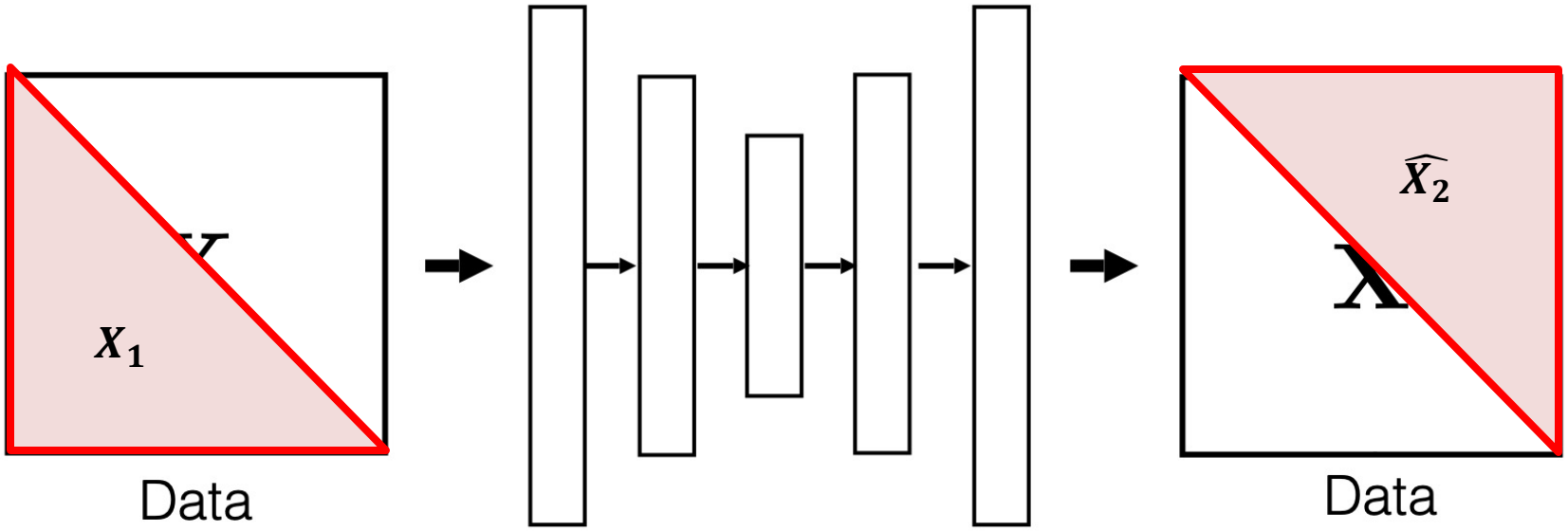
Data Compression & Task Transfer





e.g., image classification

Self-Supervision



$$F(X) = \hat{X}$$
$$F(X_1) = \hat{X}_2$$

Representation Learning

Increasing level of difficulty

Reinforcement Learning (Cherry)

Predicting a scalar reward given once in a while

A few bits for some samples

Supervised Learning (Chocolate Coat)

Predicting category or vector of scalars per input as provided by human labels.

10-10k bits per sample

Unsupervised / Self-Supervised Learning (Cake)

Predicting parts of observed input or predicting future observations or events

Millions of bits per sample

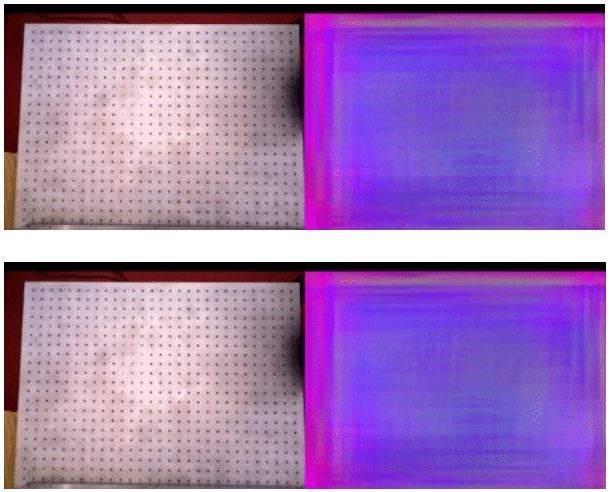
Visualisation Idea by Yann LeCun

Photo by Kristina Paukshtite from Pexels

Polleverywhere quiz

Let’s use representation learning!

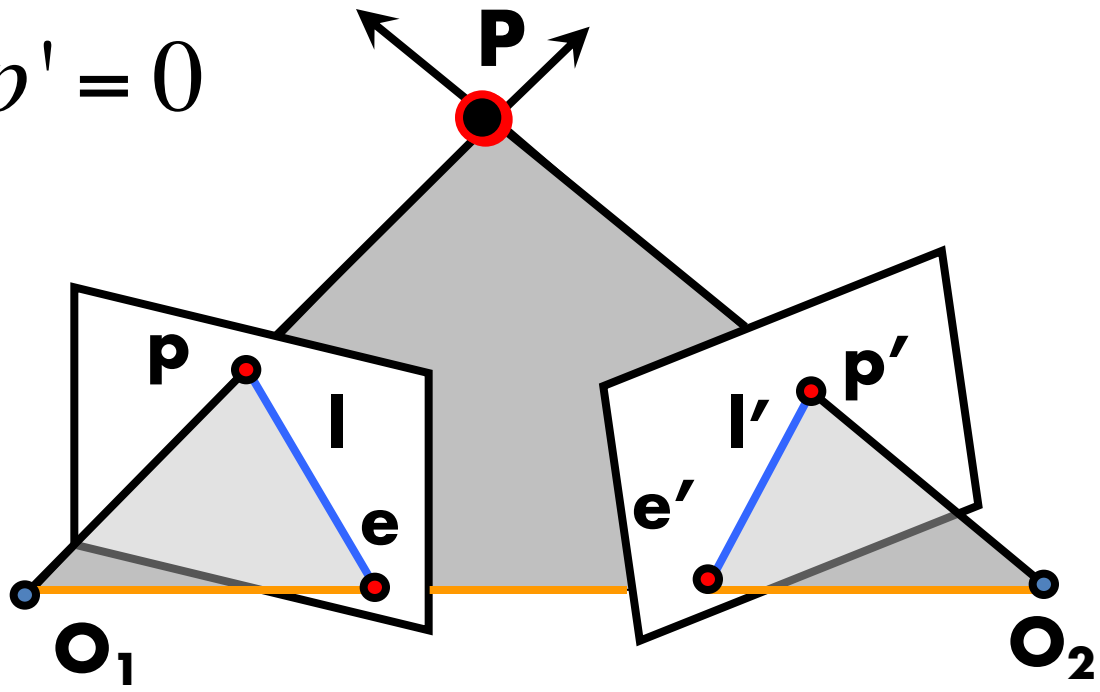
Depth Estimation



Feature Tracking

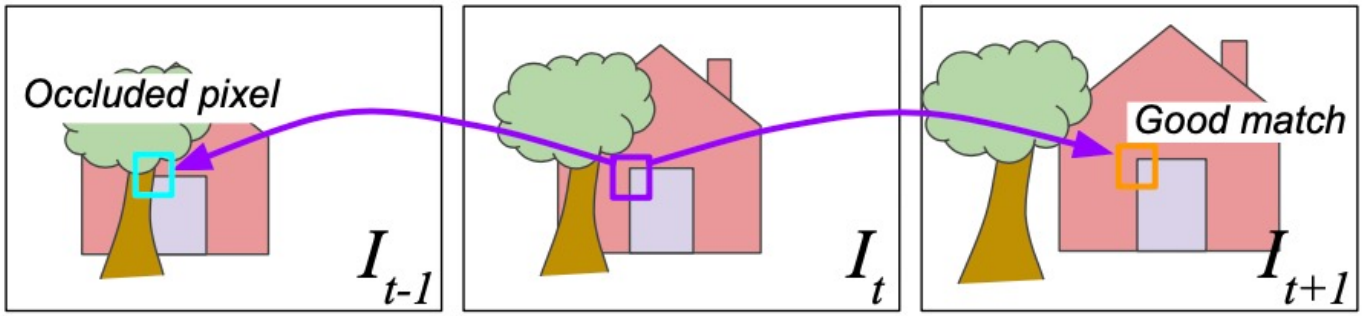
Epipolar Constraint (Lecture 5)

$$p^T \cdot F \; p' = 0$$



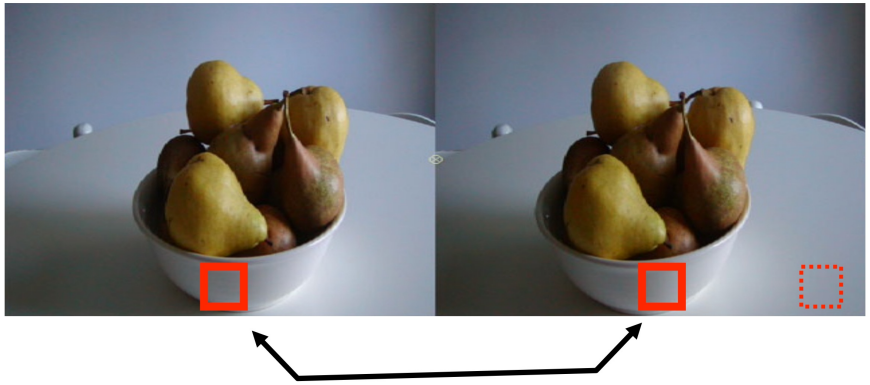
- l = F p’ is the epipolar line associated with p’
- l’= F^T p is the epipolar line associated with p
- F e’ = 0 and F^T e = 0
- F is 3x3 matrix; 7 DOF
- F is singular (rank two)

The Correspondence Problem



Occlusions

Repetitive Patterns



Homogenous regions

Can you estimate depth for a monocular image?

- Polleverywhere/breakout

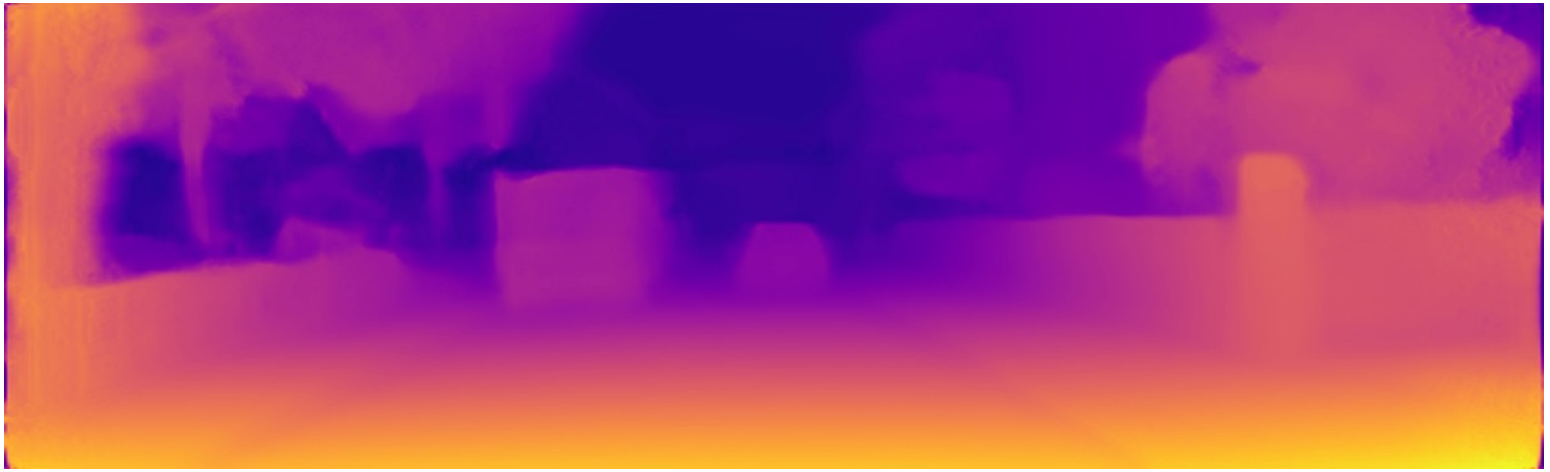
Unsupervised Monocular Depth Estimation with Left-Right Consistency

Clément Godard[1] Oisin Mac Aodha[2] Gabriel J. Brostow[1]

[1]University College London [2]Caltech

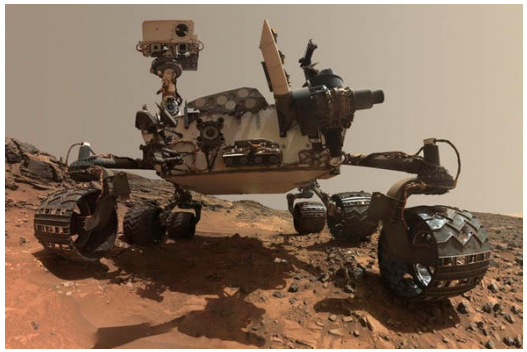
UCL

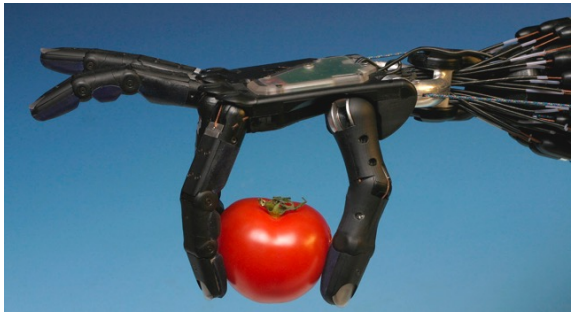
Input image



Result

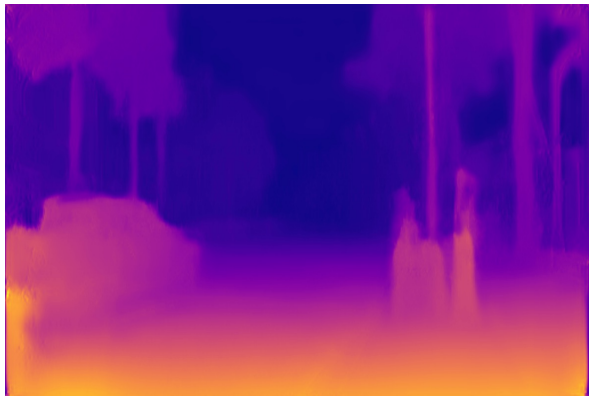
Why depth?

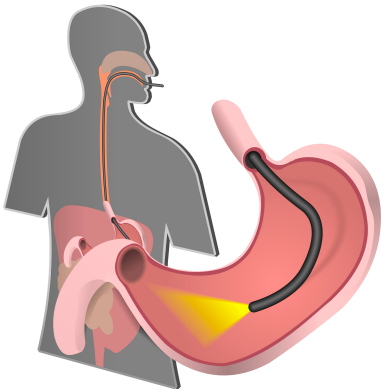




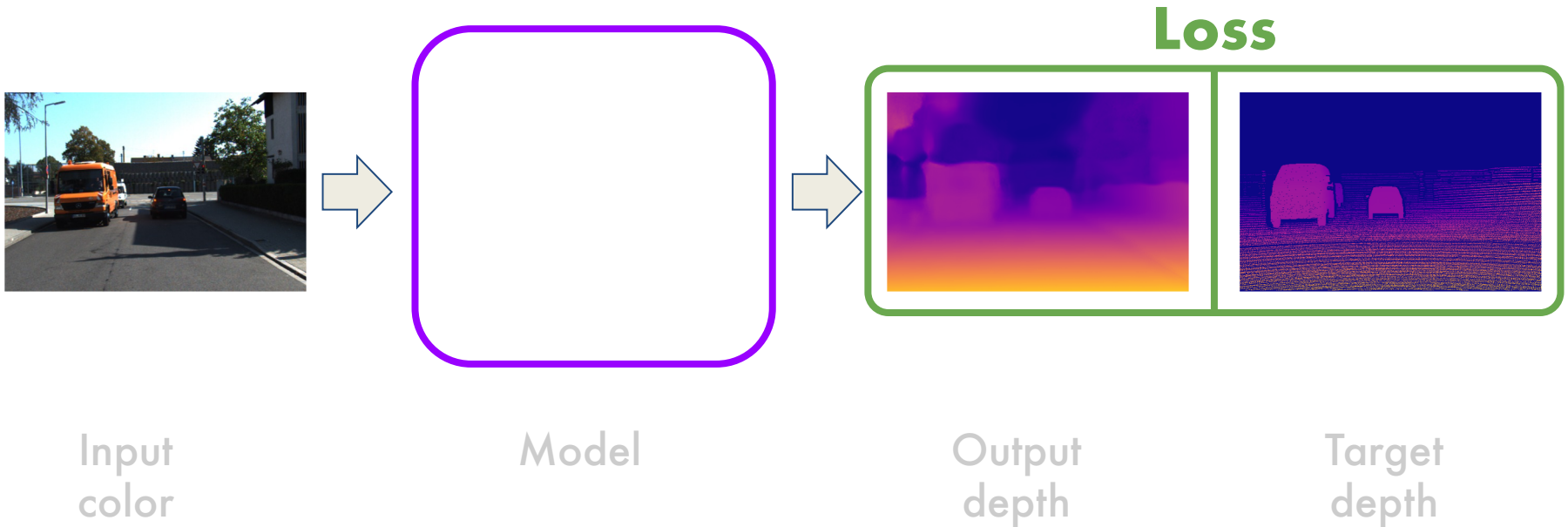
How do we usually get depth?

Why monocular?



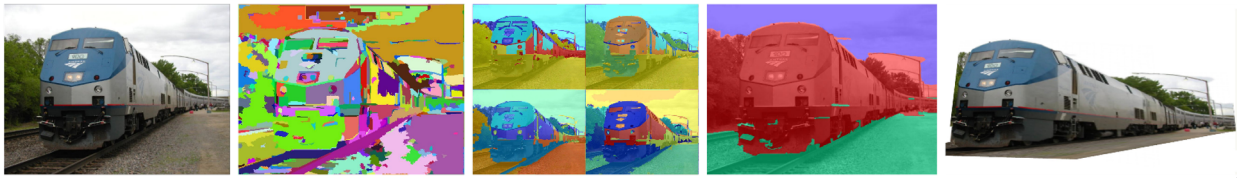


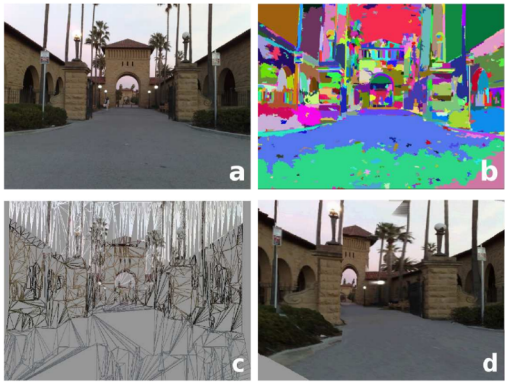
Previous approaches - Supervised



Previous approaches - Supervised

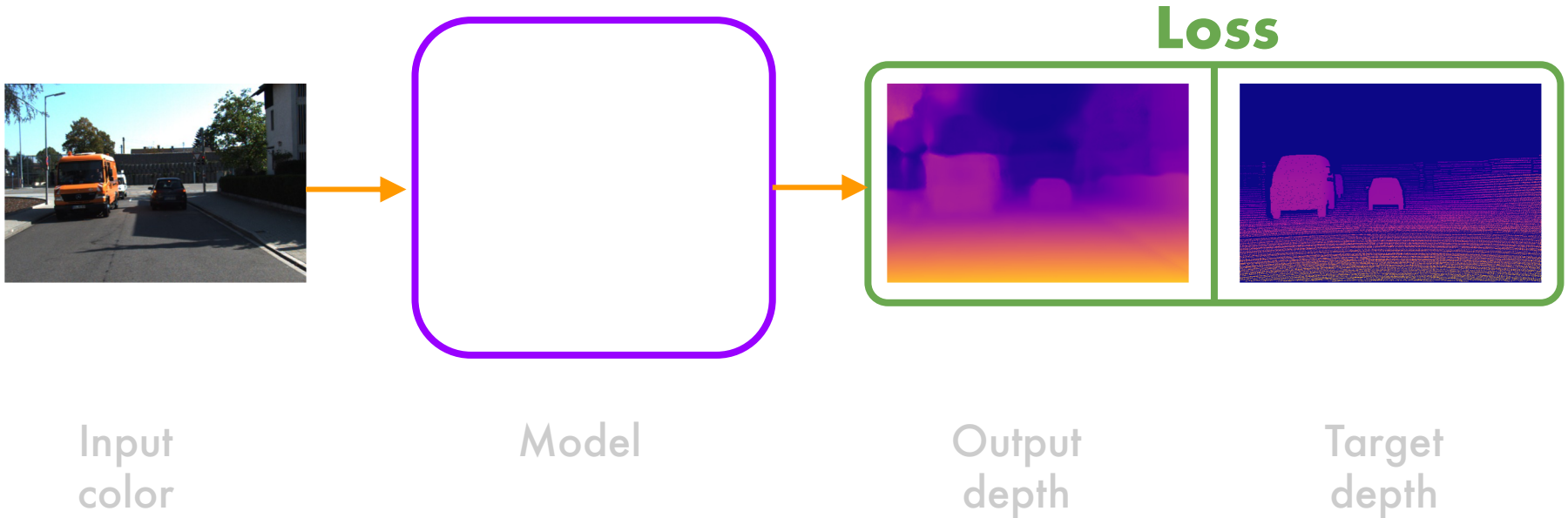
Automatic Photo Pop-up [SIGGRAPH 05]



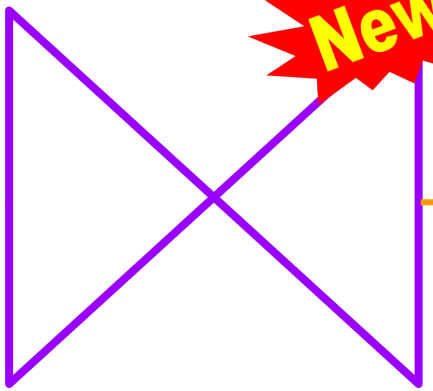


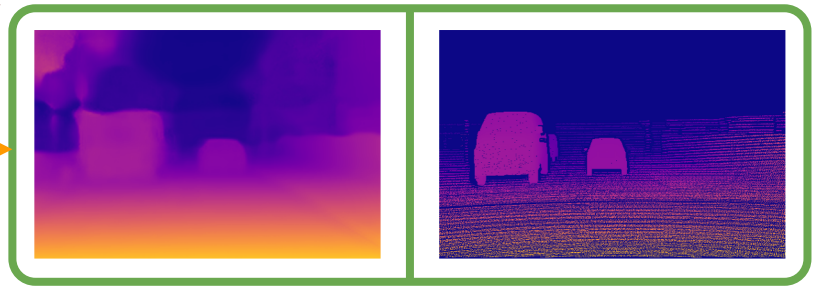
Make3D [PAMI 08]

Previous approaches - Supervised



Previous approaches - Supervised

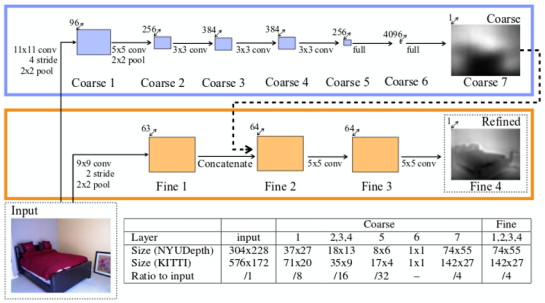




Input color

Output depth

Target depth

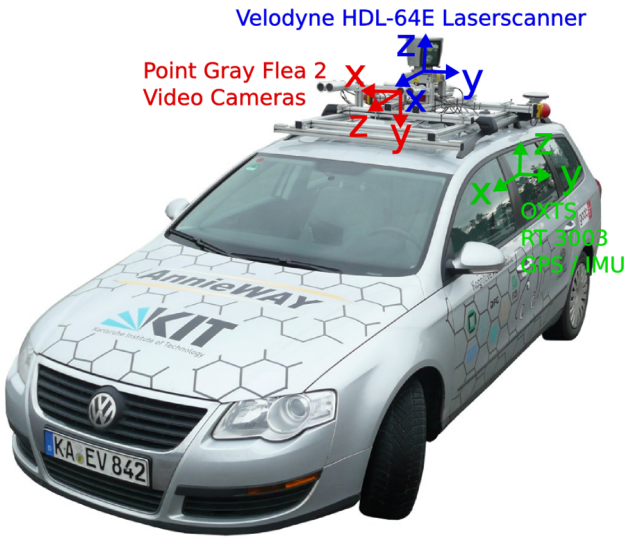


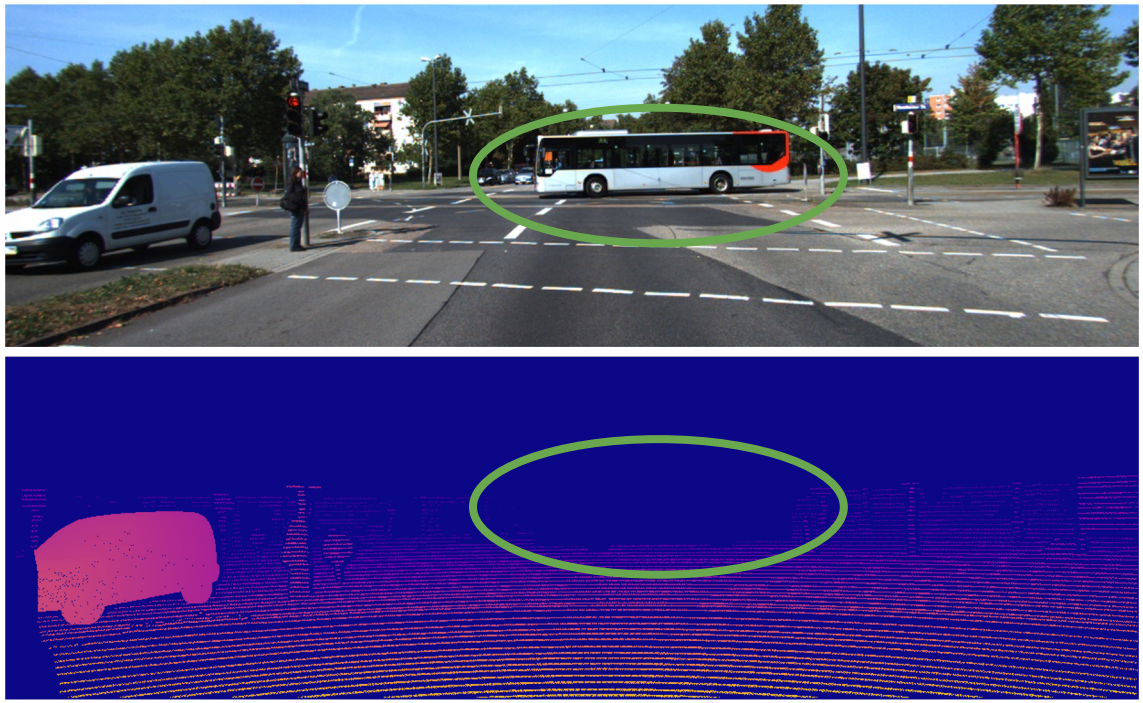
		Coarse					Fine
Layer	input	1	2,3,4	5	6	7	1,2,3,4
Size (NYUDepth)	304x228	37x27	18x13	8x6	1x1	74x55	74x55
Size (KITTI)	576x172	71x20	35x9	17x4	1x1	142x27	142x27
Ratio to input	/1	/8	/16	/32	–	/4	/4

Eigen et al. [NIPS 14]

Li et al., Laina et al., Cao et al., ...

KITTI 2015

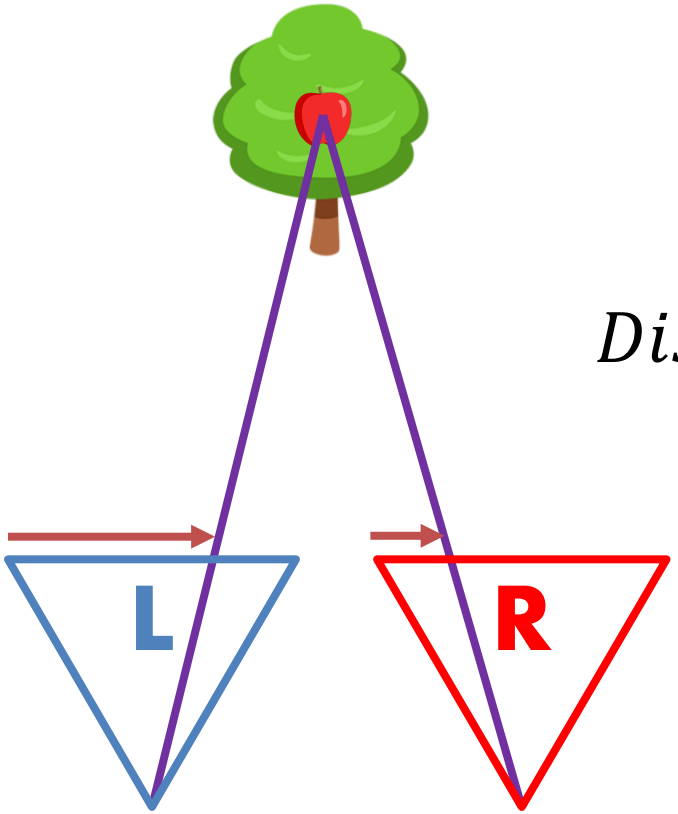

Velodyne HDL-64E Laserscanner
Point Gray Flea 2
Video Cameras
OXTS
RT 3003
GPS/IMU
KA EV 842



IR Structured Light

- Does not work well outside

Depth from stereo



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

Let’s train with stereo data!

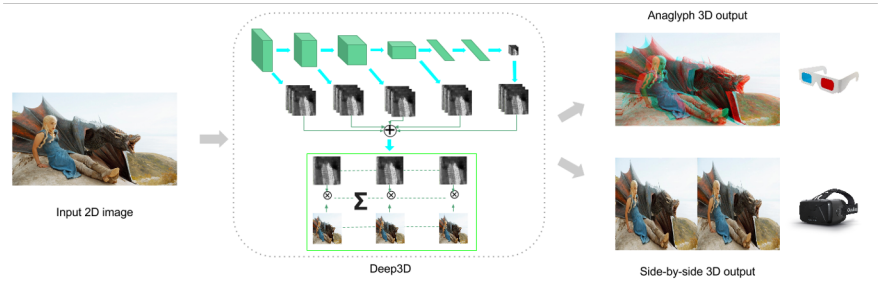
Point Grey

Apple

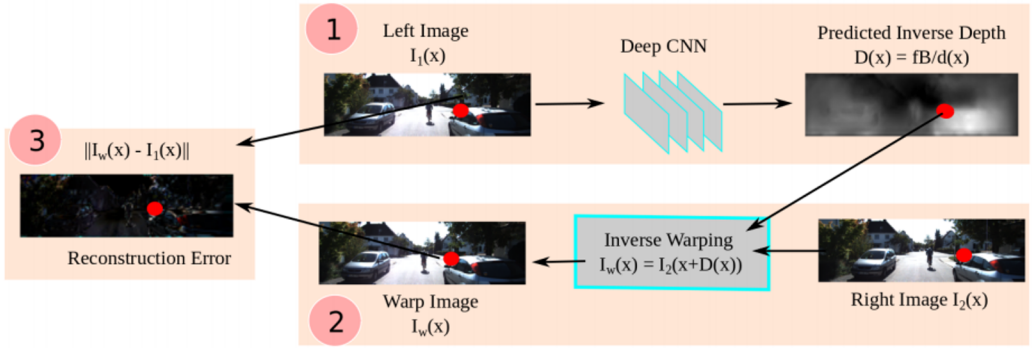
Stereolabs

Previous approaches - Unsupervised

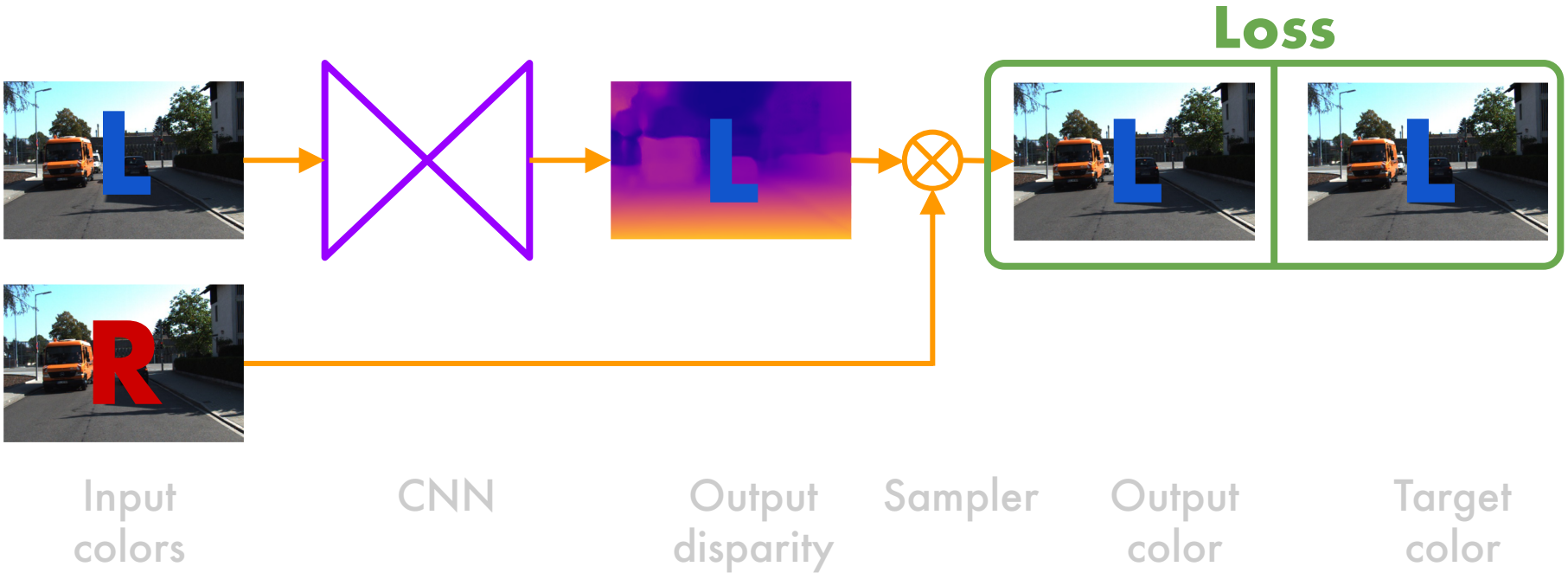
Deep3D
Xie et al. [ECCV 16]



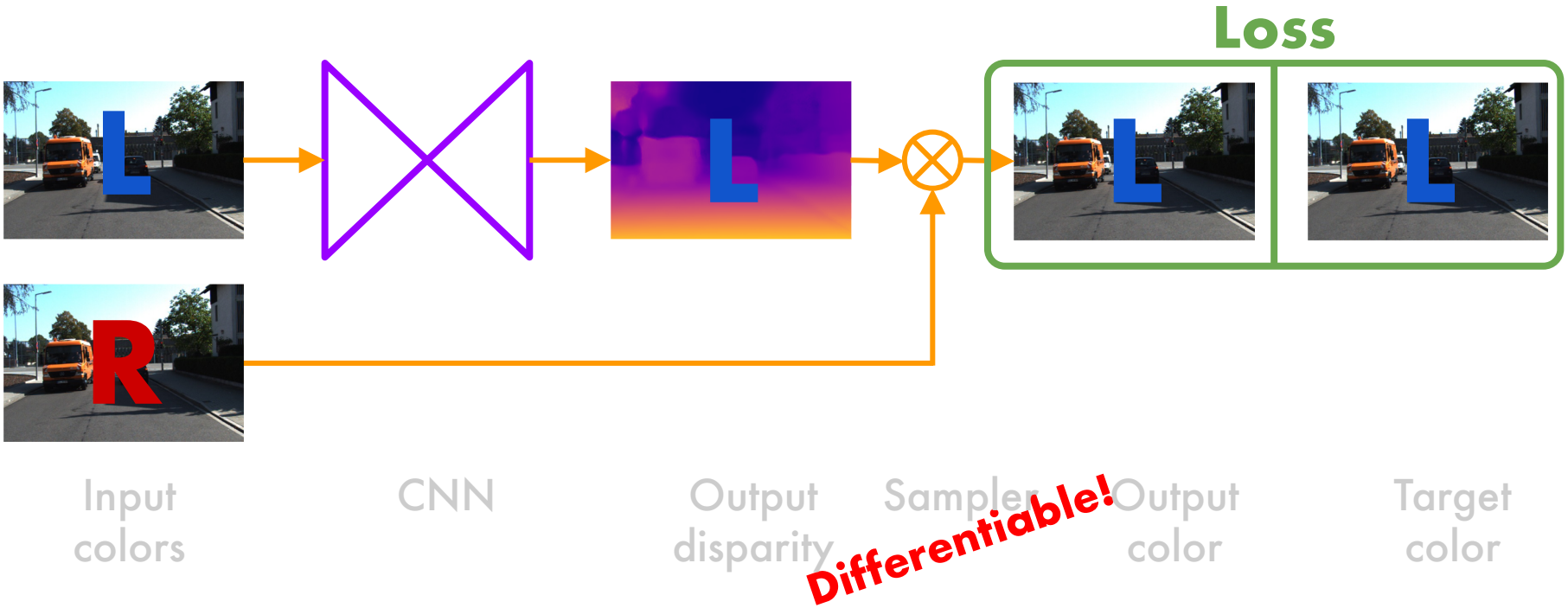
Garg et al. [ECCV 16]



Unsupervised depth estimation - Concept

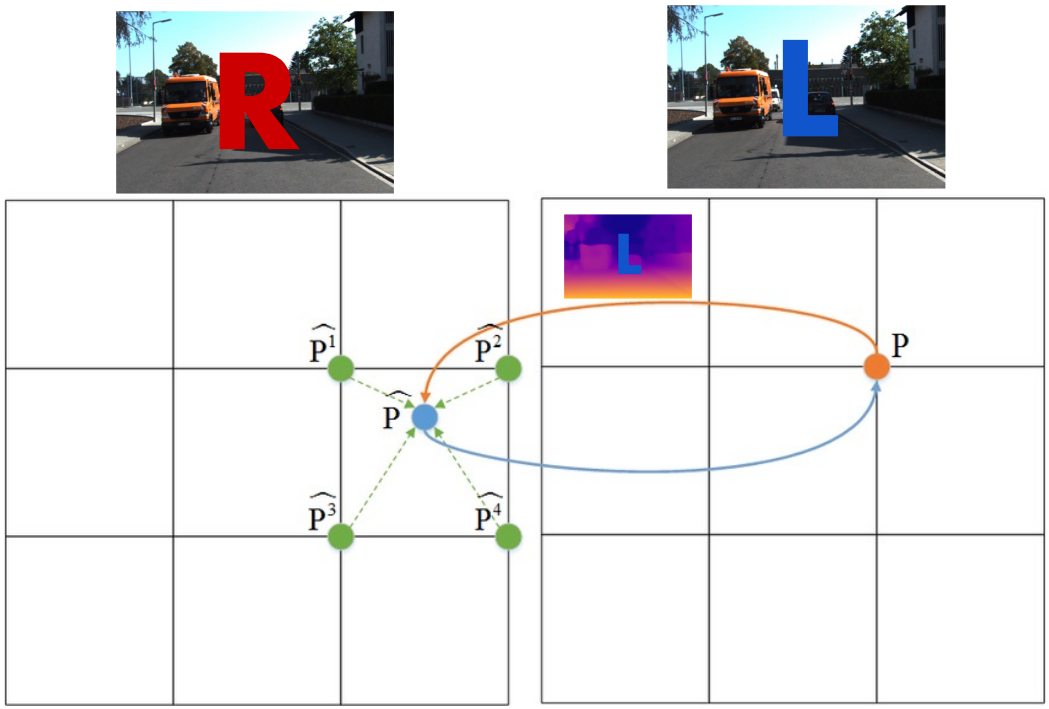


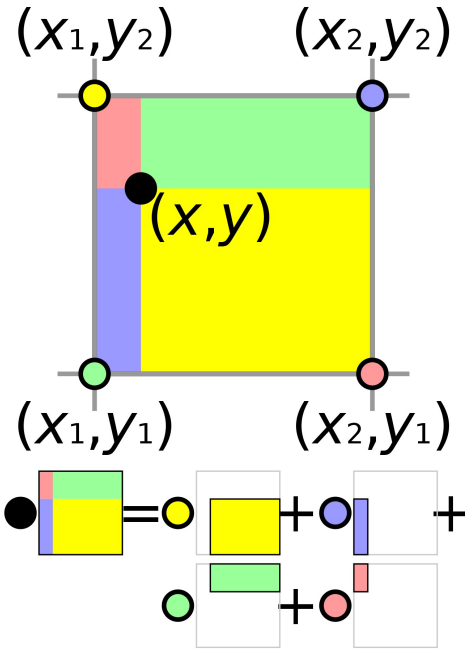
Unsupervised depth estimation - Baseline



Spatial transformer networks, Jaderberg et al. [NIPS 15]

Bilinear Sampling



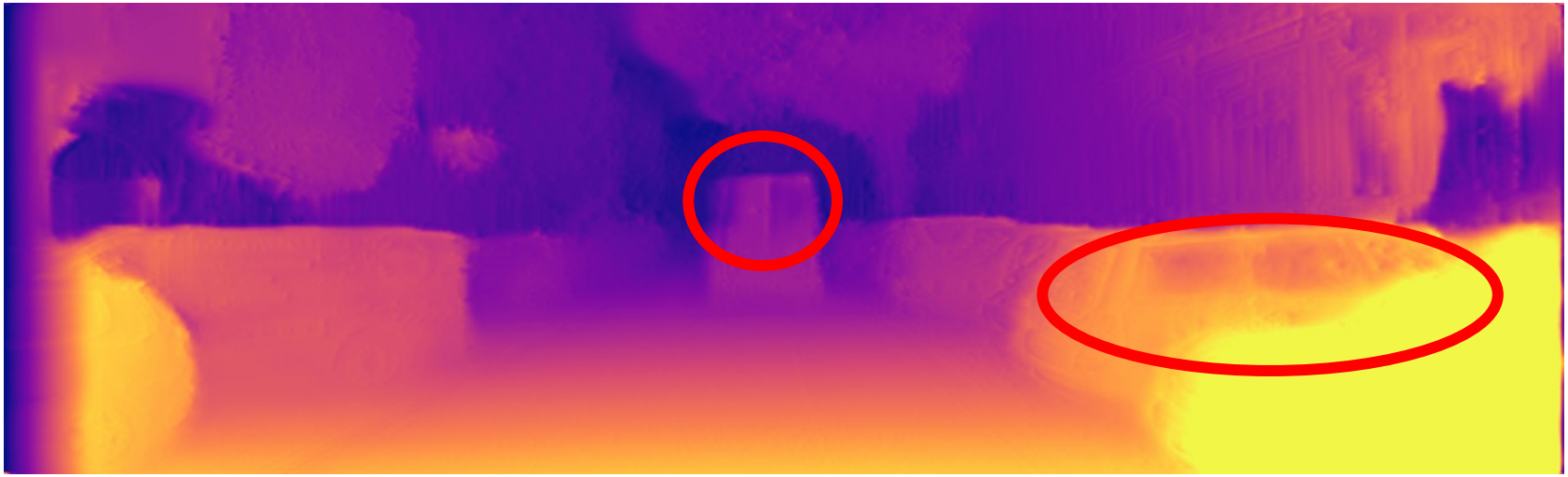


Spatial transformer networks, Jaderberg et al. [NIPS 15]

Input

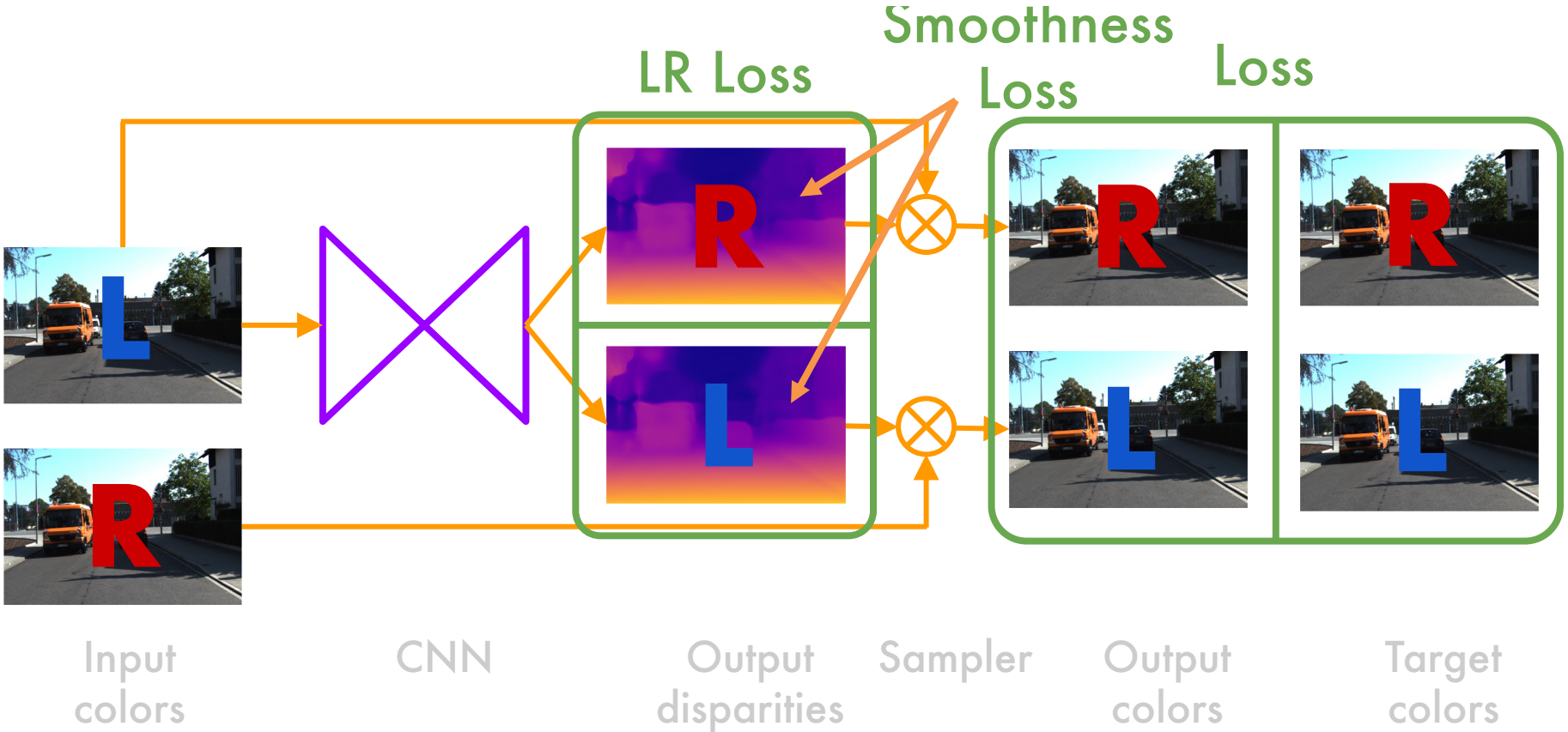
Baseline

This method



Unsupervised depth estimation - This method

$$C_s = \alpha_{ap}(C_{ap}^l + C_{ap}^r) + \alpha_{ds}(C_{ds}^l + C_{ds}^r) + \alpha_{lr}(C_{lr}^l + C_{lr}^r)$$



Reconstruction Loss

Complete Loss $C_s = \alpha_{ap}(C_{ap}^l + C_{ap}^r) + \alpha_{ds}(C_{ds}^l + C_{ds}^r) + \alpha_{lr}(C_{lr}^l + C_{lr}^r)$

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

Luminance Contrast Structure

SSIM: Structural Similarity Index = $f(l(\mathbf{x},\mathbf{y}), c(\mathbf{x},\mathbf{y}), s(\mathbf{x},\mathbf{y}))$

Left-Right Consistency Loss

Complete Loss $C_s = \alpha_{ap}(C_{ap}^l + C_{ap}^r) + \alpha_{ds}(C_{ds}^l + C_{ds}^r) + \alpha_{lr}(C_{lr}^l + C_{lr}^r)$

$$C_{lr}^l = \frac{1}{N} \sum_{i,j} \left| d_{ij}^l - d_{ij+d_{ij}^l}^r \right|$$

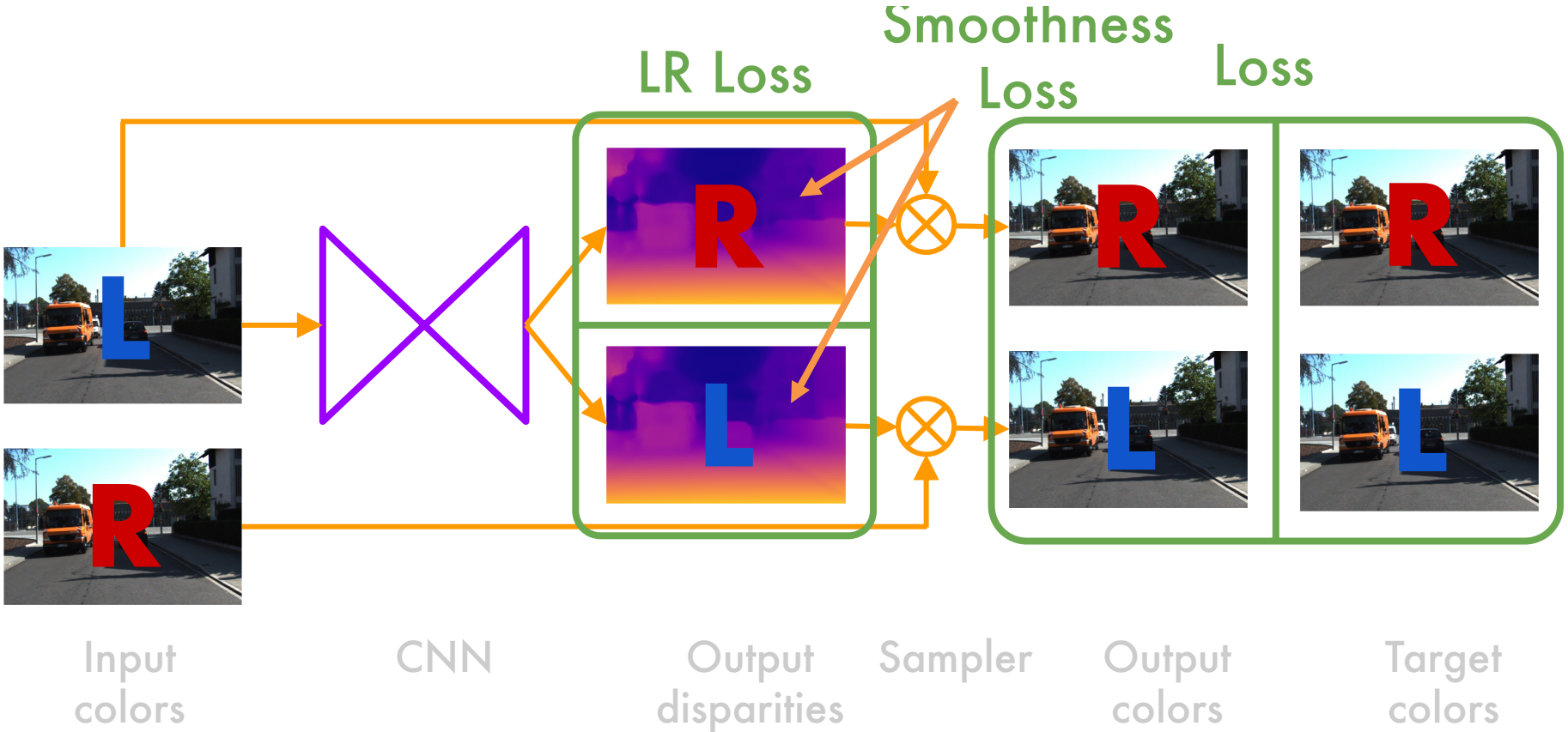
Smoothness Loss

Complete Loss $C_s = \alpha_{ap}(C_{ap}^l + C_{ap}^r) + \alpha_{ds}(C_{ds}^l + C_{ds}^r) + \alpha_{lr}(C_{lr}^l + C_{lr}^r)$

$$C_{ds}^l = \frac{1}{N}\sum_{i,j} |\partial_x d_{ij}^l| e^{-\|\partial_x I_{ij}^l\|} + |\partial_y d_{ij}^l| e^{-\|\partial_y I_{ij}^l\|}$$

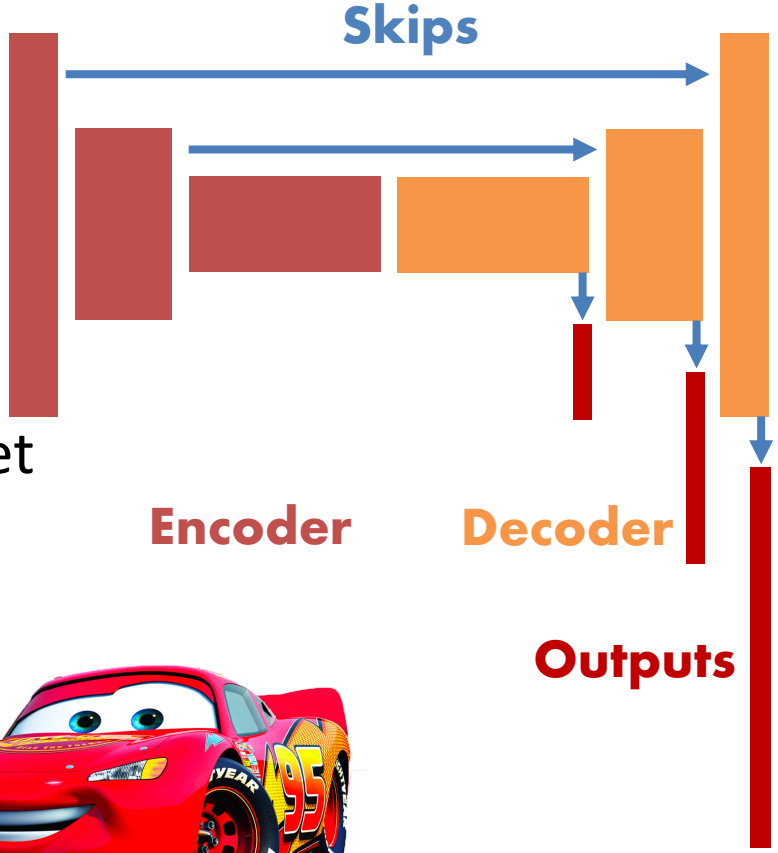
Unsupervised depth estimation - This method

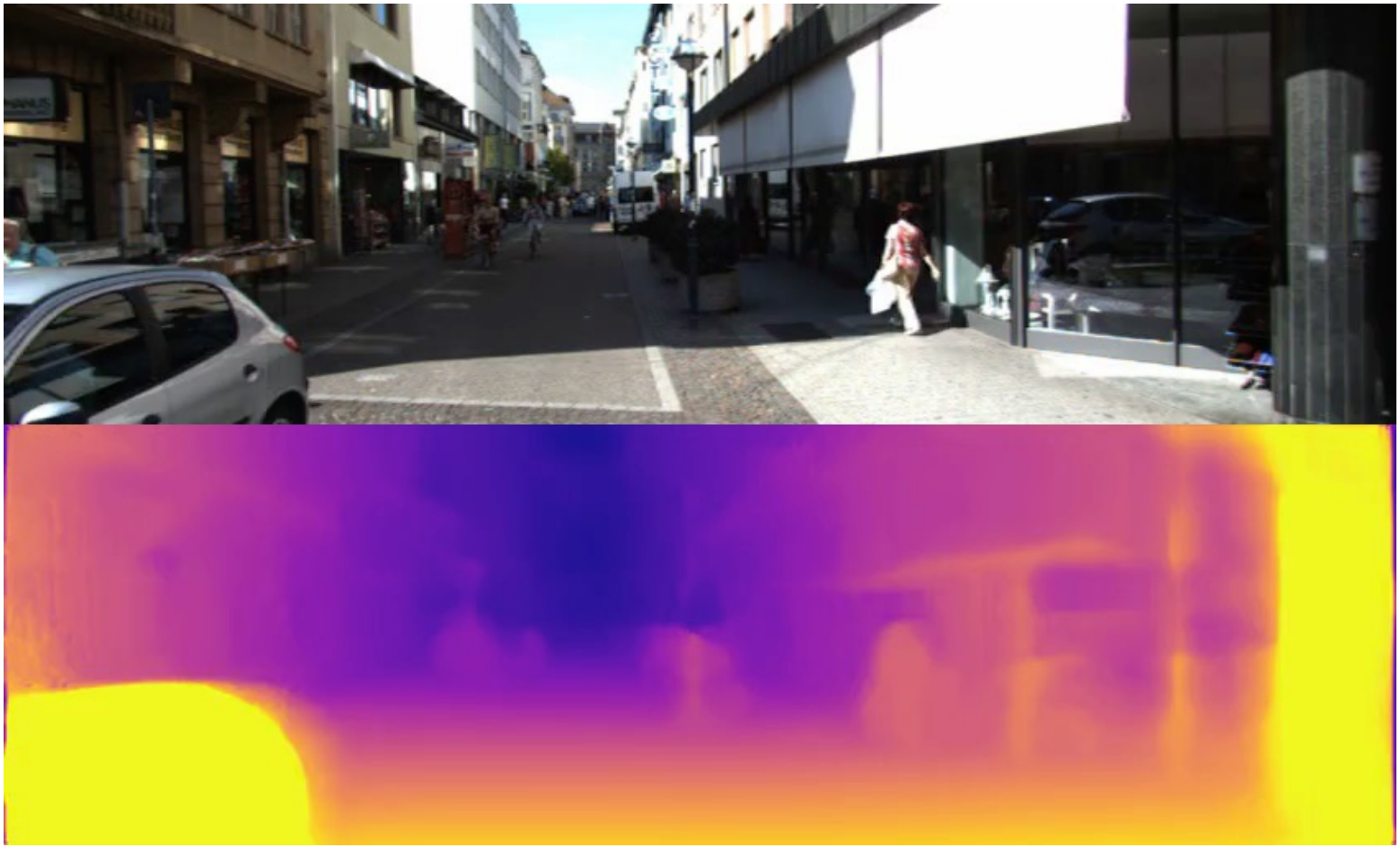
$$C_s = \alpha_{ap}(C_{ap}^l + C_{ap}^r) + \alpha_{ds}(C_{ds}^l + C_{ds}^r) + \alpha_{lr}(C_{lr}^l + C_{lr}^r)$$



Architecture

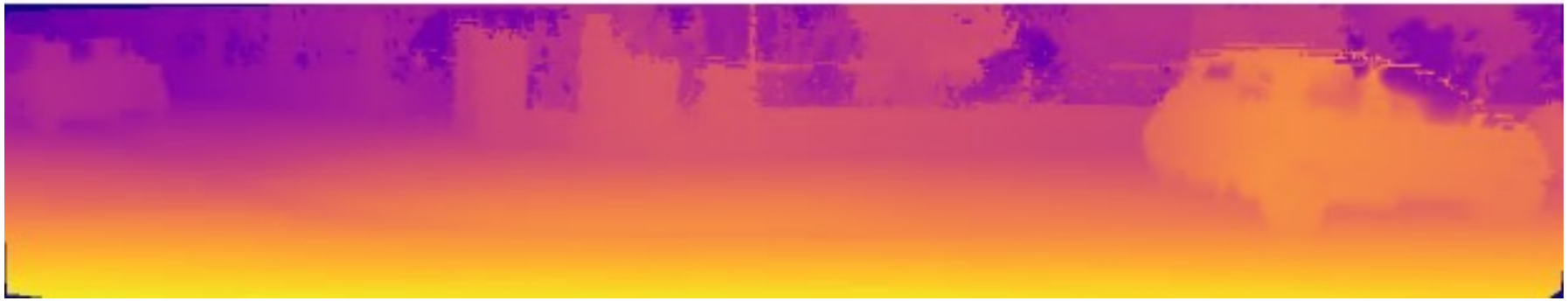
- Fully convolutional
  - Choose your favorite encoder
- Skip connections
  - Similar to DispNet and FlowNet
- Multiscale generation
  - And Loss!
- Fast!
  - ~30fps on a Titan X



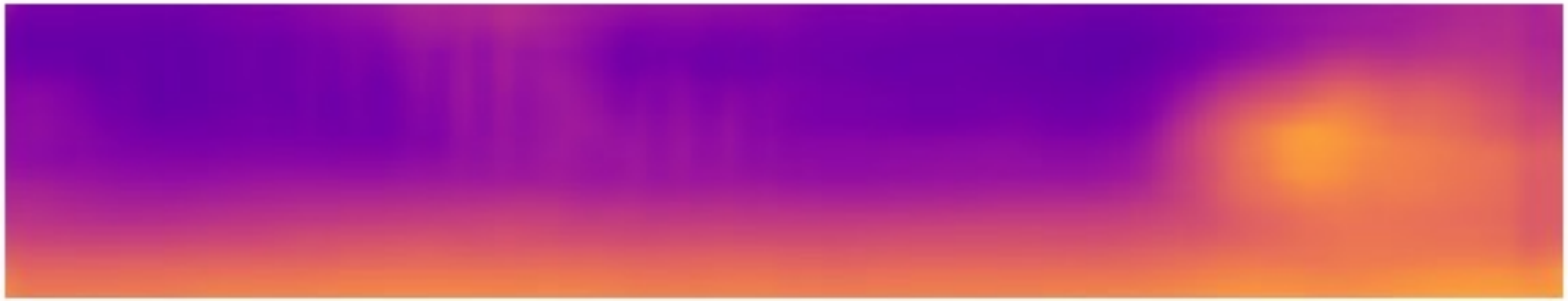


KITTI – Input image

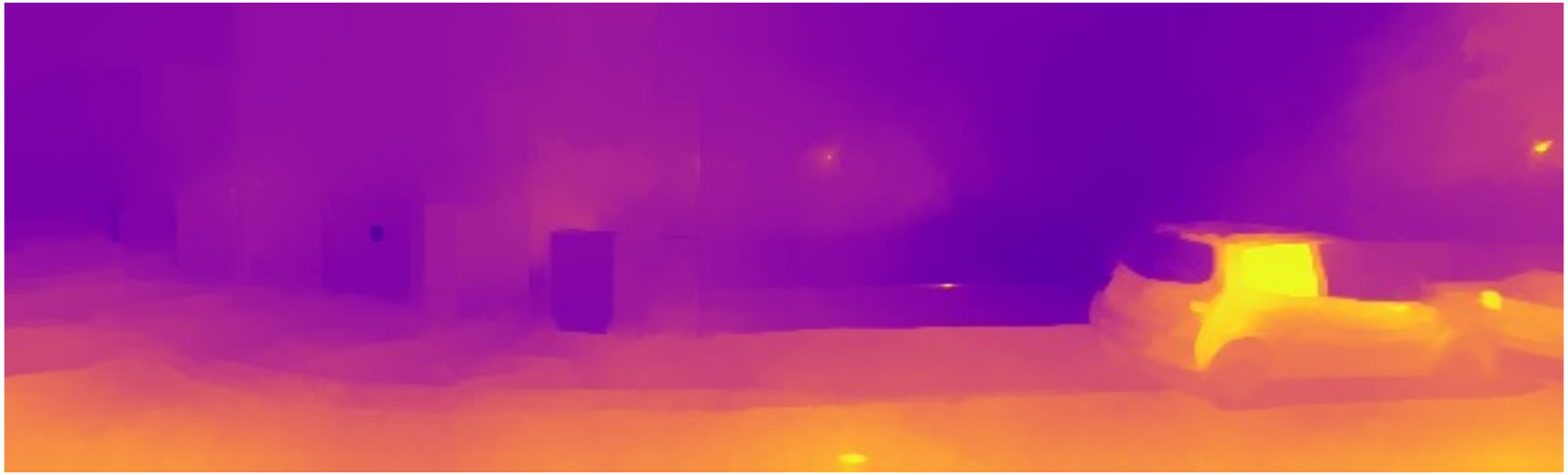
KITTI – Ground truth depth



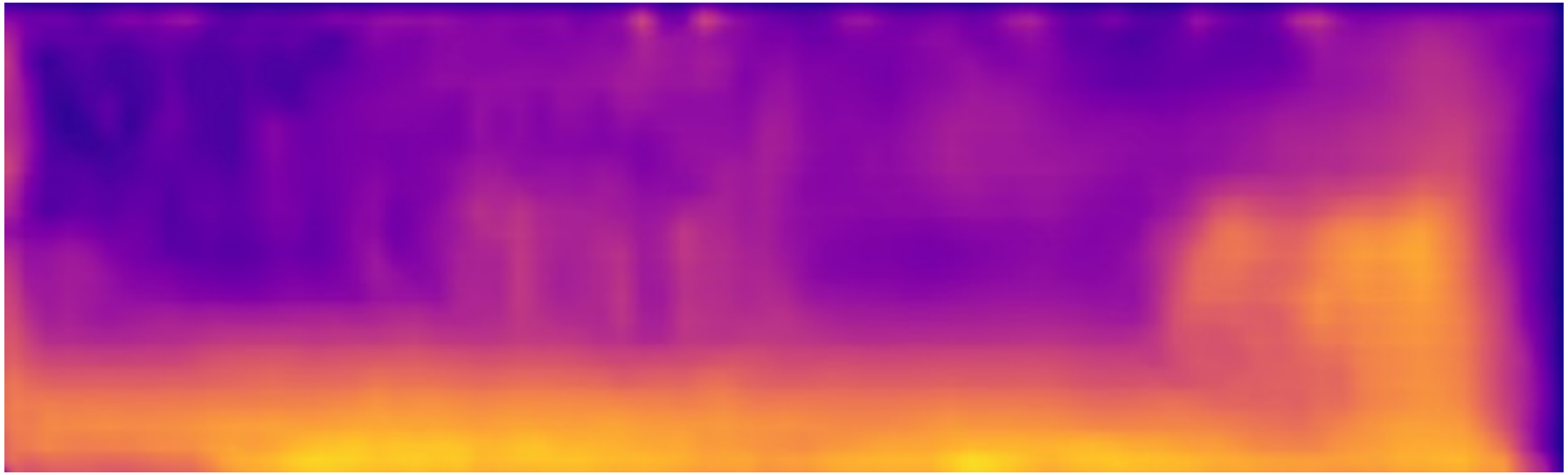
KITTI – Eigen *et al.* [NIPS 14]



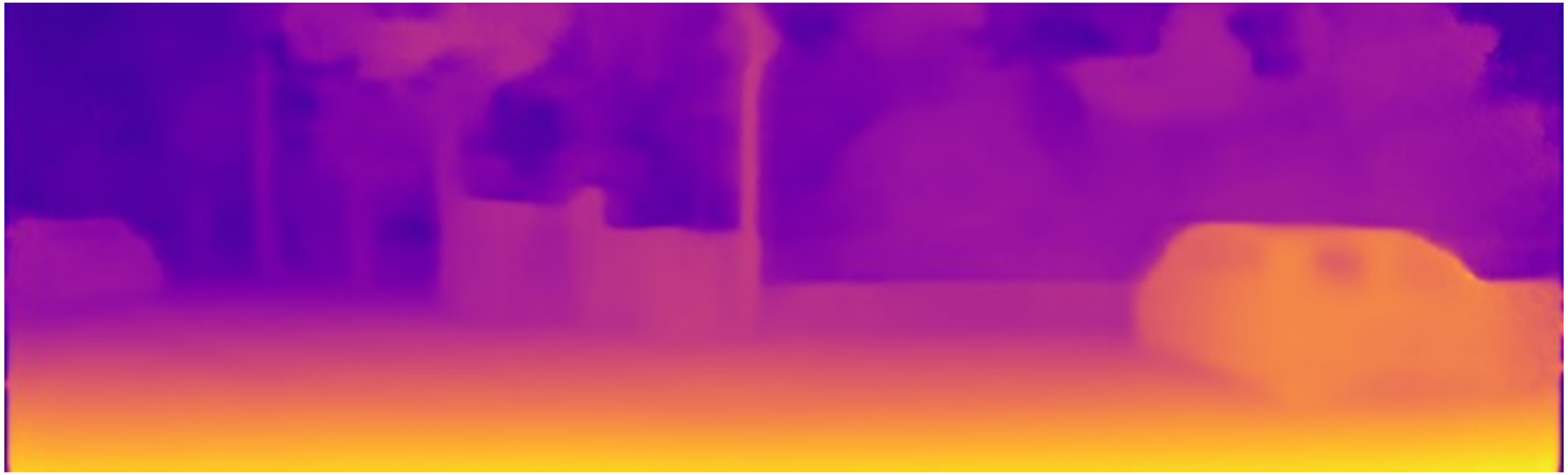
KITTI – Liu *et al.* [CVPR 14]



KITTI – Garg *et al.* [ECCV 16]



KITTI – This work



KITTI – Input image

KITTI 2015

- All variants of our model beat previous supervised methods

Method	Supervised	Dataset	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Train set mean	No	K	0.361	4.826	8.102	0.377	0.638	0.804	0.894
Eigen et al. [10] Coarse °	Yes	K	0.214	1.605	6.563	0.292	0.673	0.884	0.957
Eigen et al. [10] Fine °	Yes	K	0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu et al. [36] DCNF-FCSP FT *	Yes	K	0.201	1.584	6.471	0.273	0.68	0.898	0.967
Ours No LR	No	K	0.152	1.528	6.098	0.252	0.801	0.922	0.963
Ours	No	K	0.148	1.344	5.927	0.247	0.803	0.922	0.964
Ours	No	CS + K	0.124	1.076	5.311	0.219	0.847	0.942	0.973
Ours pp	No	CS + K	0.118	0.923	5.015	0.210	0.854	0.947	**0.976**
Ours resnet pp	No	CS + K	**0.114**	**0.898**	**4.935**	**0.206**	**0.861**	**0.949**	**0.976**

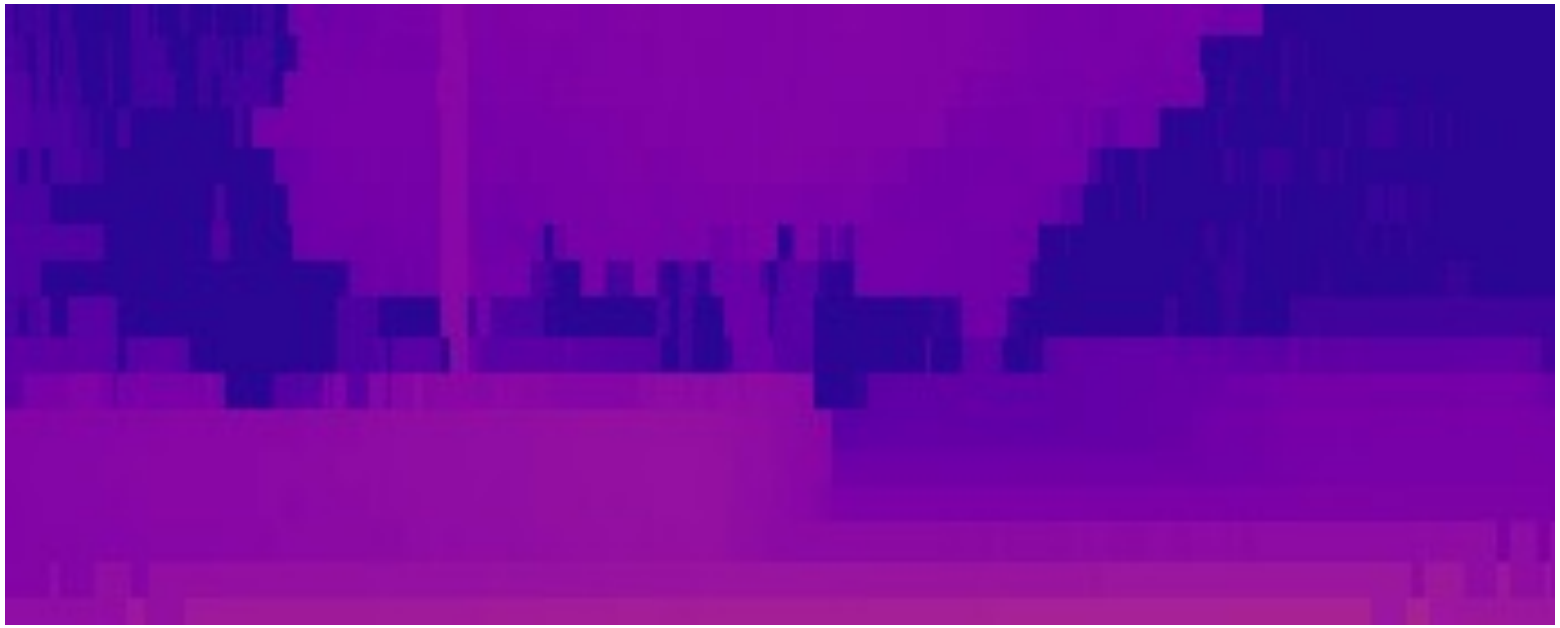
KITTI 2015

- All variants of our model beat the previous unsupervised method

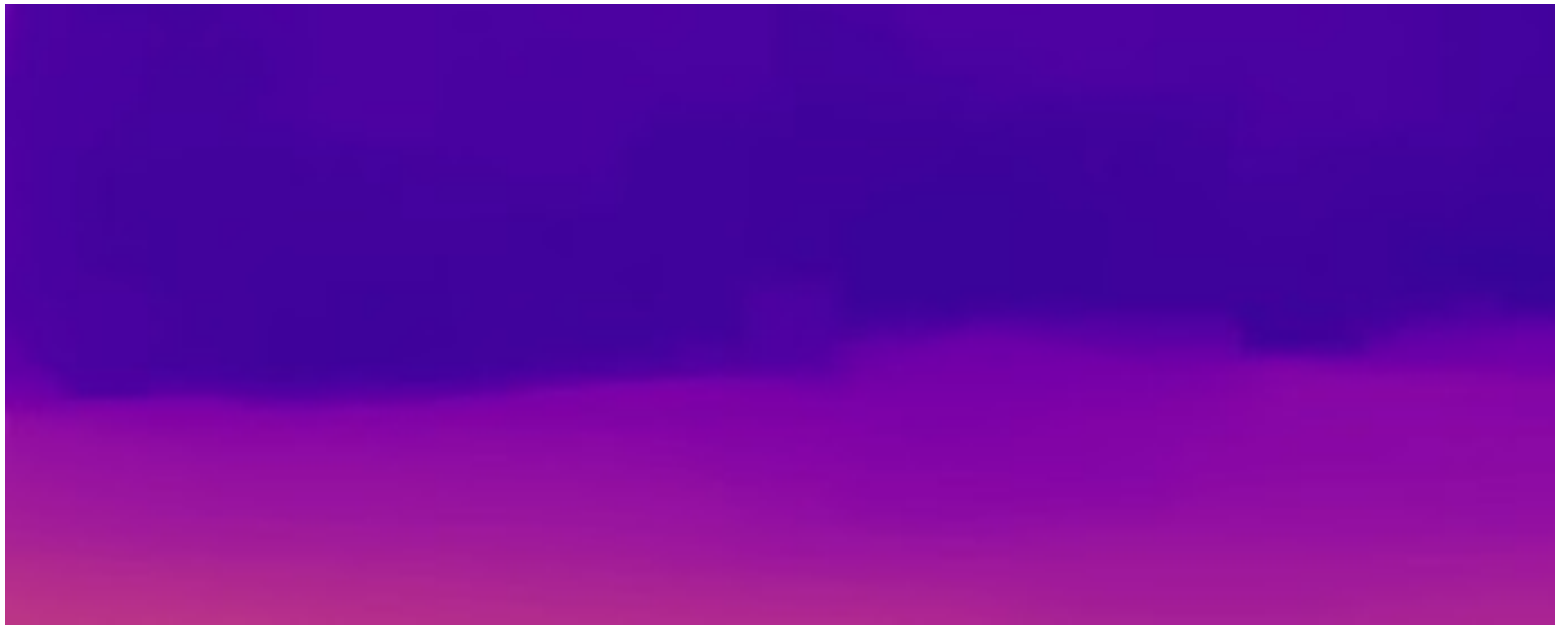
Method	Supervised	Dataset	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Train set mean	No	K	0.361	4.826	8.102	0.377	0.638	0.804	0.894
Eigen et al. [10] Coarse °	Yes	K	0.214	1.605	6.563	0.292	0.673	0.884	0.957
Eigen et al. [10] Fine °	Yes	K	0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu et al. [36] DCNF-FCSP FT *	Yes	K	0.201	1.584	6.471	0.273	0.68	0.898	0.967
Ours No LR	No	K	0.152	1.528	6.098	0.252	0.801	0.922	0.963
Ours	No	K	0.148	1.344	5.927	0.247	0.803	0.922	0.964
Ours	No	CS + K	0.124	1.076	5.311	0.219	0.847	0.942	0.973
Ours pp	No	CS + K	0.118	0.923	5.015	0.210	0.854	0.947	**0.976**
Ours resnet pp	No	CS + K	**0.114**	**0.898**	**4.935**	**0.206**	**0.861**	**0.949**	**0.976**
Garg et al. [16] L12 Aug 8× cap 50m	No	K	0.169	1.080	5.104	0.273	0.740	0.904	0.962
Ours cap 50m	No	K	0.140	0.976	4.471	0.232	0.818	0.931	0.969
Ours cap 50m	No	CS + K	0.117	0.762	3.972	0.206	0.860	0.948	0.976
Ours pp cap 50m	No	CS + K	0.112	0.680	3.810	0.198	0.866	0.953	**0.979**
Ours resnet pp cap 50m	No	CS + K	**0.108**	**0.657**	**3.729**	**0.194**	**0.873**	**0.954**	**0.979**

Make3D – Input image

Make3D – Ground truth depth

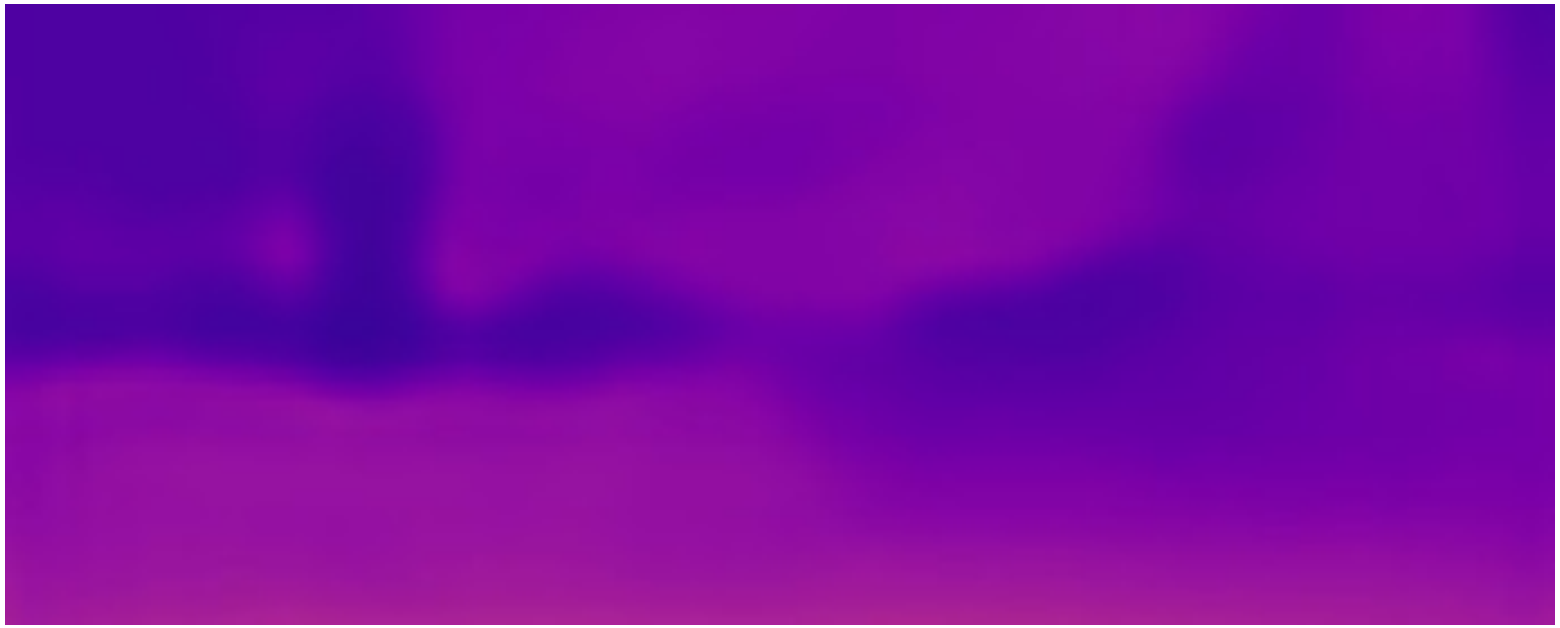


Make3D – Karsch *et al.* [PAMI 14]

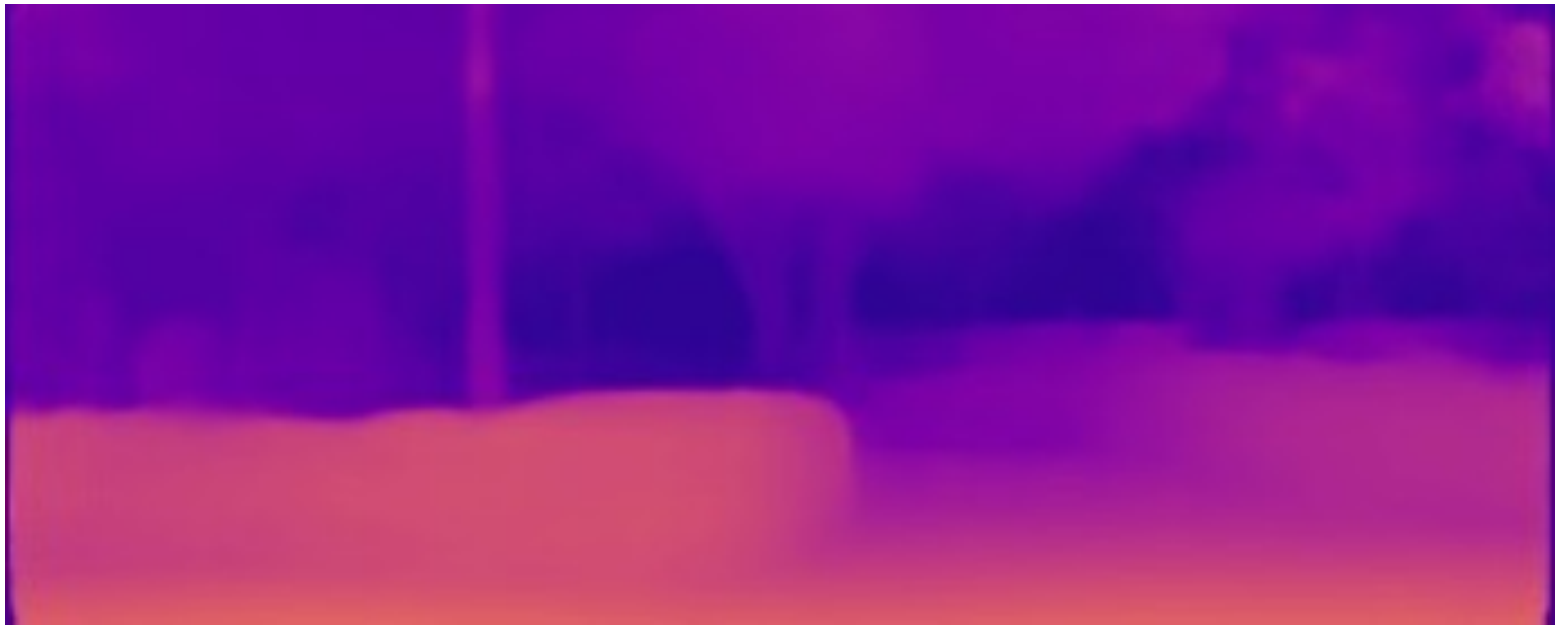


Make3D – Liu *et al.* [CVPR 14]

Make3D – Laina *et al.* [3DV 16]



Make3D – This method



Make3D – Input image

Challenges

- Reprojection loss
  - Assumes lambertian world
  - More supervision?
    - Kuznietsov *et al.* [CVPR 17]

- Need calibrated data
  - Synced and rectified
  - Less supervision?
    - Zhou *et al.* [CVPR 2017]

Conclusion

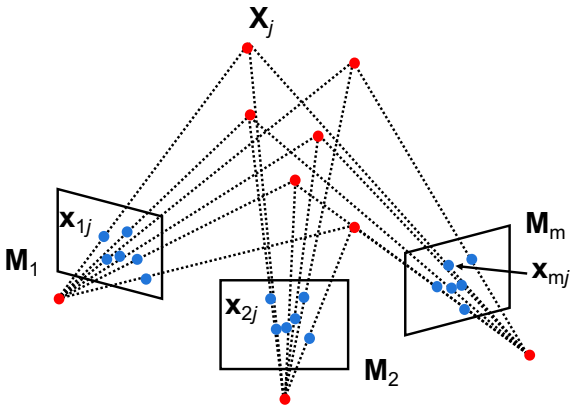
- We can get depth from a single photograph
- Self-supervision with stereo data
  - Cheap and scalable!
- Accurate
  - Beats fully-supervised methods on KITTI!

Digging into Self-Supervised Monocular Depth Estimation

Clément Godard[1] Oisin Mac Aodha[2] Michael Firman[3] Gabriel J. Brostow[1]

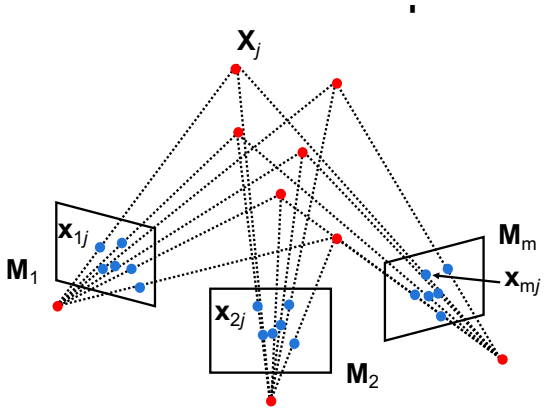
[1]University College London [2]Caltech [3]Niantic

Structure From Motion Problem



Given *m* images of *n* fixed 3D points

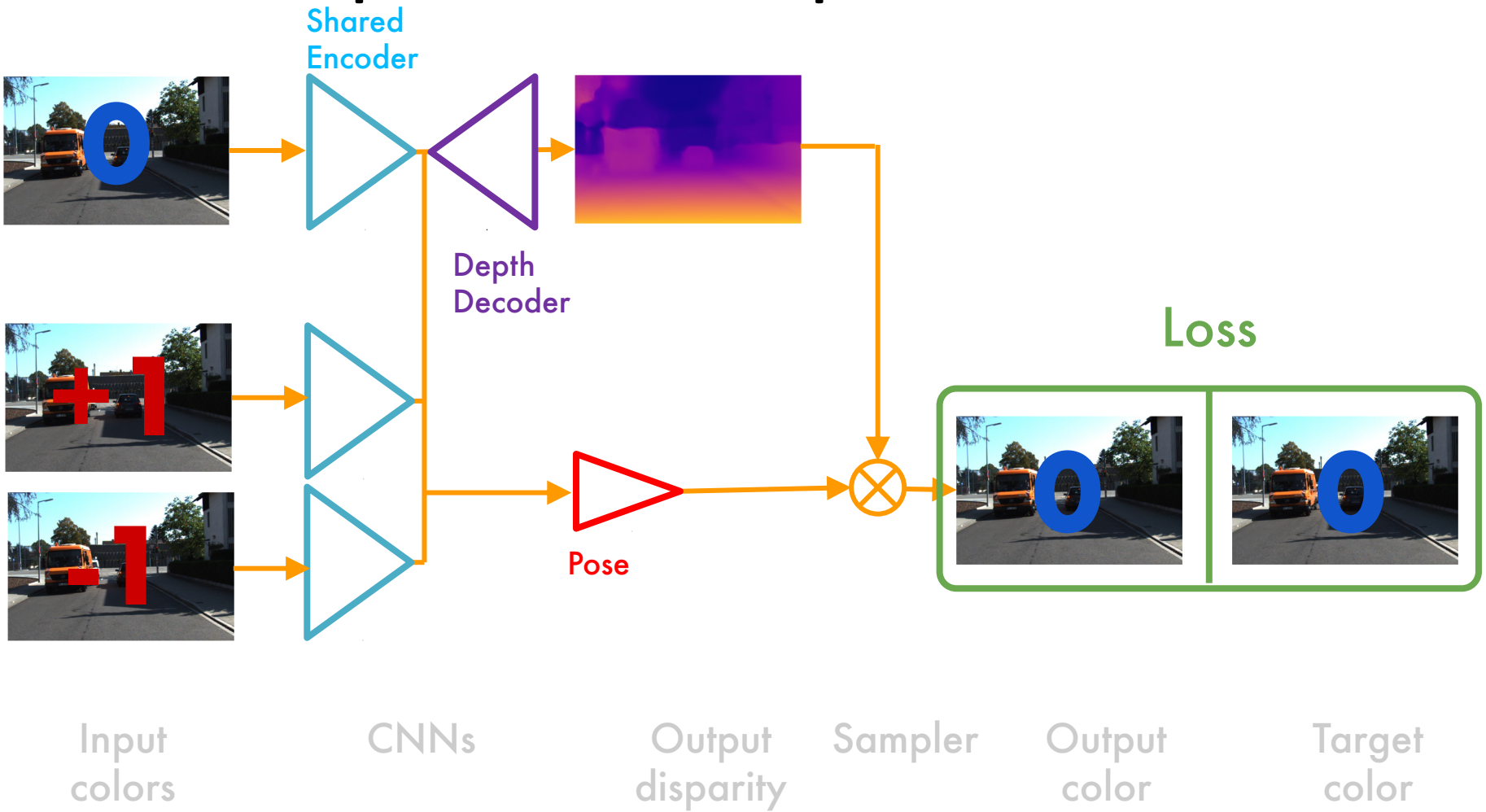
- $\mathbf{x}_{ij} = \mathbf{M}_i \mathbf{X}_j, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n$



From the mxn observations $\mathbf{x}_{ij}$, estimate:

- *m* projection matrices $\mathbf{M}_i$ — motion
- *n* 3D points $\mathbf{X}_j$ — structure

Self-Supervised Depth Estimation



Loss functions

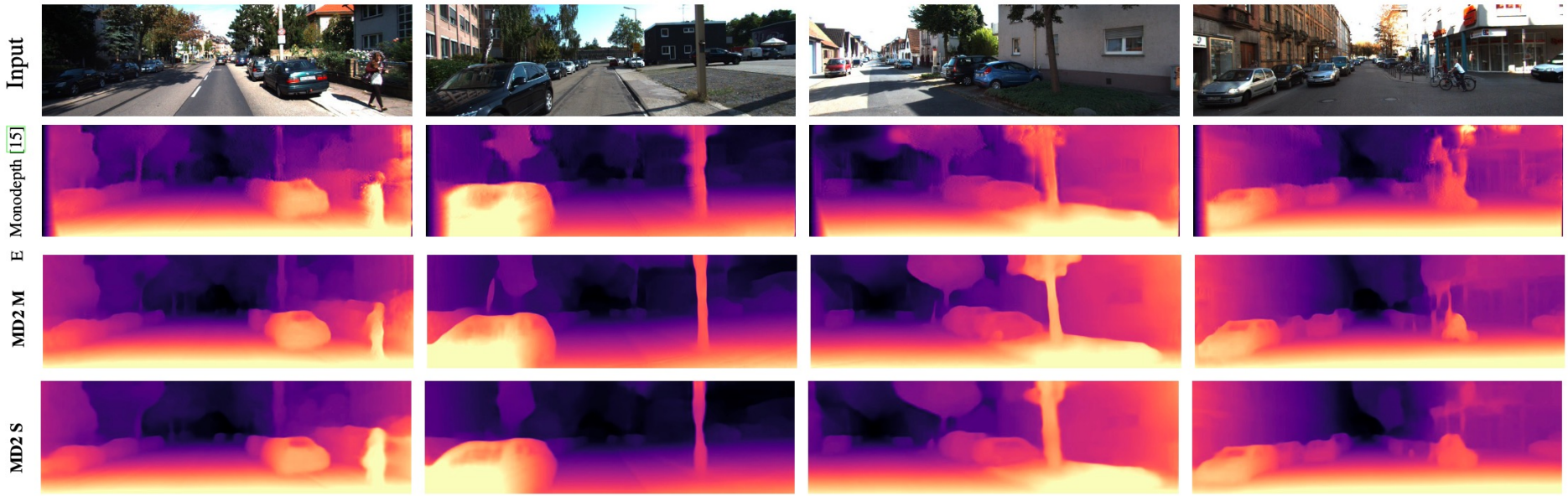
$$L_p = \sum_{t'} pe(I_t, I_{t' \to t}),$$

$$I_{t' \to t} = I_{t'} \left\langle proj(D_t, T_{t \to t'}, K) \right\rangle$$

$$pe(I_a, I_b) = \frac{\alpha}{2}(1 - \text{SSIM}(I_a, I_b)) + (1 - \alpha)\|I_a - I_b\|_1$$

$$L_s = |\partial_x d_t^*| \, e^{-|\partial_x I_t|} + |\partial_y d_t^*| \, e^{-|\partial_y I_t|}$$

Results

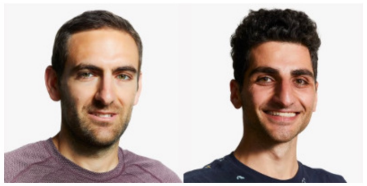


Examples in Industry

Robotics Today - A Series of Technical Talks

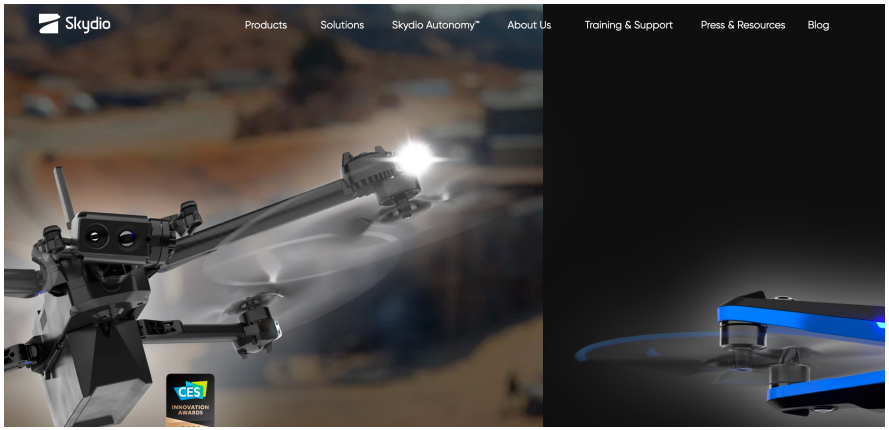
Watch | Upcoming Seminars | Past Seminars | Organizers | Contact

"Robotics Today - A series of technical talks" is a virtual robotics seminar series. The goal of the series is to bring the robotics community together during these challenging times. **The seminars are scheduled on Fridays at 3PM EST (12AM PST) and are open to the public**. The format of the seminar consists of a technical talk live captioned and streamed via Web and Twitter (@RoboticsSeminar), followed by an interactive discussion between the speaker and a panel of faculty, postdocs, and students that will moderate audience questions.



Skydio Autonomy: Research in Robust Visual Navigation and Real-Time 3D Reconstruction
12 February 2021: Adam Bry (Skydio) and Hayk Martiros (Skydio)

Abstract: Skydio is the leading US drone company and the world leader in autonomous flight. Our drones are used for everything from capturing amazing video, to inspecting bridges, to tracking progress on construction sites. At the core of our products is a vision-based autonomy system with seven years of development at Skydio, drawing on decades of academic research. This system pushes the state of the art in deep learning, geometric computer vision, motion planning, and control with a particular focus on real-world robustness. Drones encounter extreme visual scenarios not typically considered by academia nor encountered by cars, ground robots, or AR applications. They are commonly flown in scenes with few or no semantic priors and must deftly navigate thin objects, extreme lighting, camera artifacts, motion blur, textureless surfaces, vibrations, dirt, camera smudges, and fog. These challenges are daunting for classical vision - because photometric signals are simply not consistent and for learning-based methods - because there is no ground truth for direct supervision of deep networks. In this talk we'll take a detailed look at these issues and the algorithms we've developed to tackle them. We will also cover the new capabilities on top of our core navigation engine to autonomously map complex scenes and capture all surfaces, by performing real-time 3D reconstruction across multiple flights.



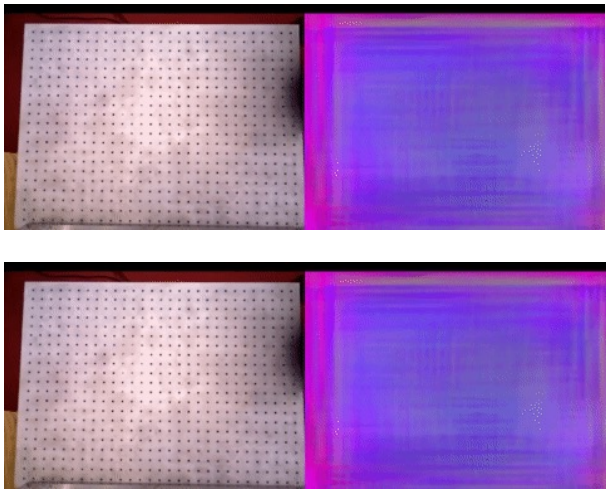
Toyota Research Institute

Teaching Robot to help People in their home.

TOYOTA

Let's use representation learning!

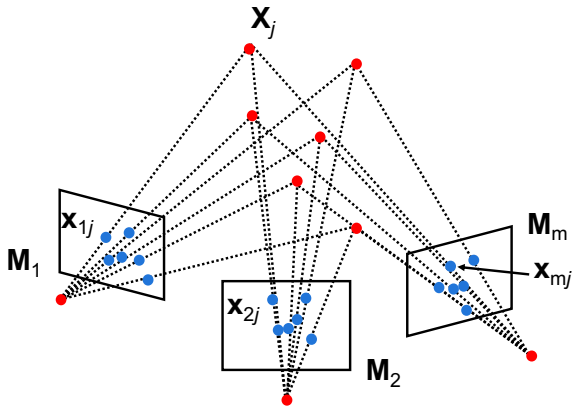
Depth Estimation



Feature Tracking

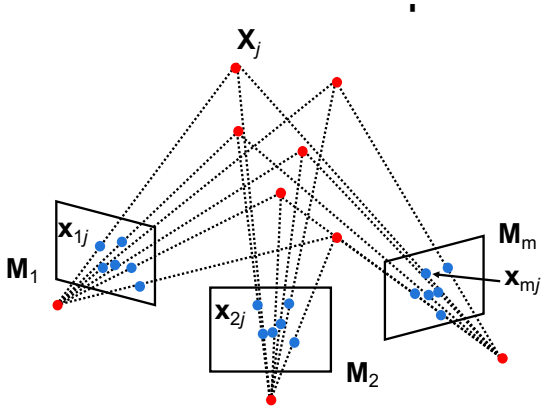
Feature Tracking

Structure From Motion Problem



Given *m* images of *n* fixed 3D points

- $\mathbf{x}_{ij} = \mathbf{M}_i \mathbf{X}_j, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n$



From the m×n observations $\mathbf{x}_{ij}$, estimate:

- *m* projection matrices $\mathbf{M}_i$ — motion
- *n* 3D points $\mathbf{X}_j$ — structure

Problem statement

Image sequence

Slide credit: Yonsei Univ.

Slides Adapted from CS131a.

Problem statement

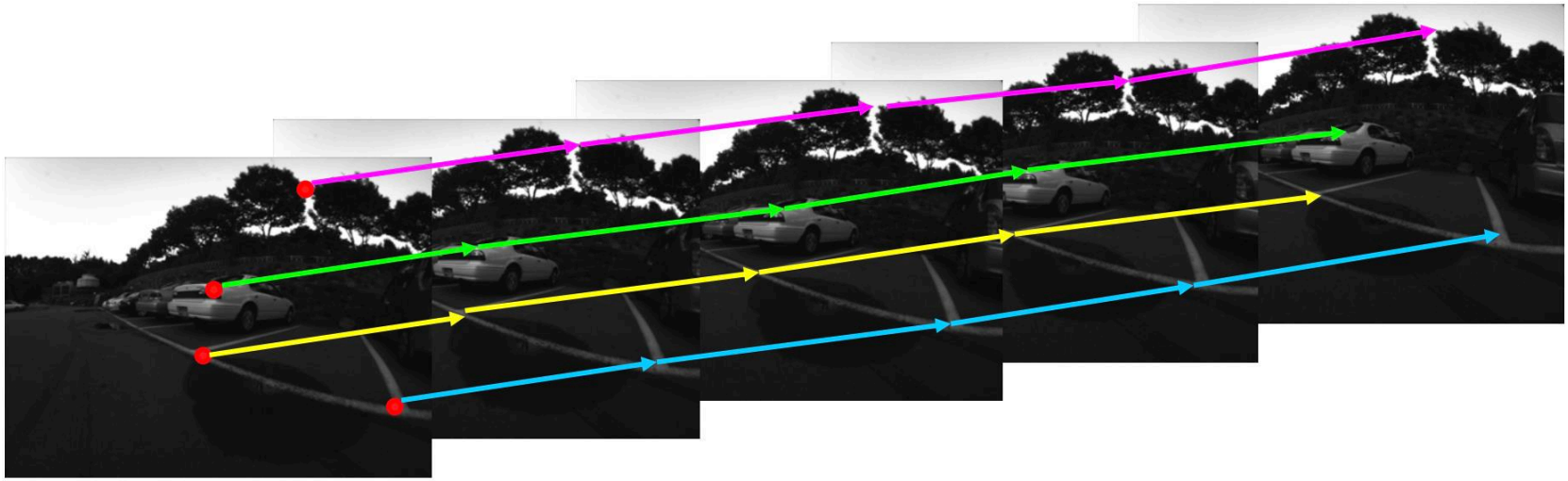
Feature point detection

Slide credit: Yonsei Univ.

Slides Adapted from CS131a.

Problem statement

Feature point tracking



Slide credit: Yonsei Univ.

Slides Adapted from CS131a.

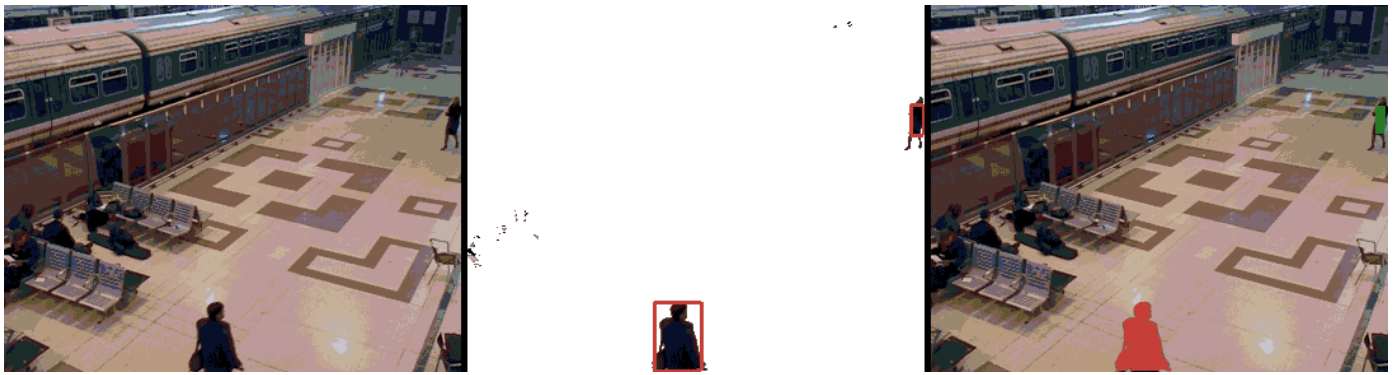
Single object tracking

Slides Adapted from CS131a.

Multiple object tracking

Slides Adapted from CS131a.

Tracking with a fixed camera



Slides Adapted from CS131a.

Tracking with a moving camera



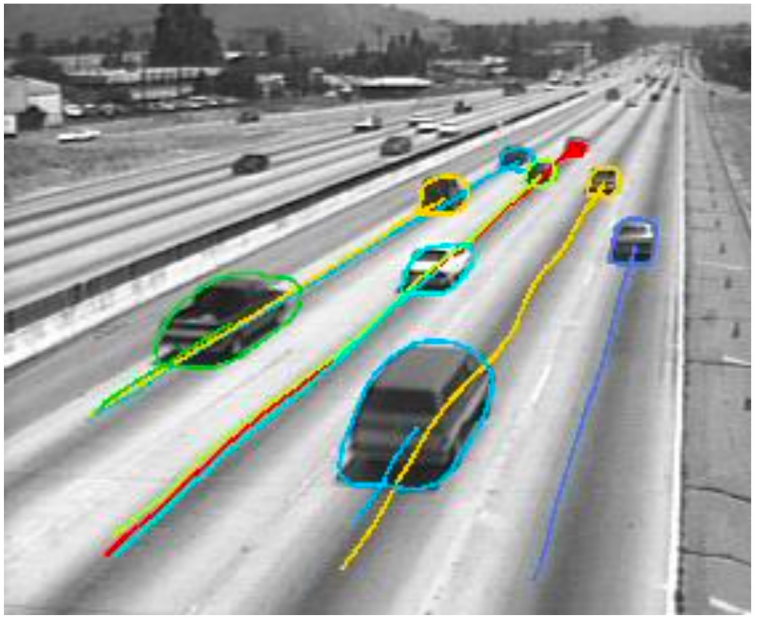
Slides Adapted from CS131a.

Challenges in Feature Tracking

- Figure out which features can be tracked
  - Efficiently track across frames
- Some points may change appearance over time
  - e.g., due to rotation, moving into shadows, etc.
- Drift: small errors can accumulate as appearance model is updated
- Points may appear or disappear.
  - need to be able to add/delete tracked points.

What are good features to track?

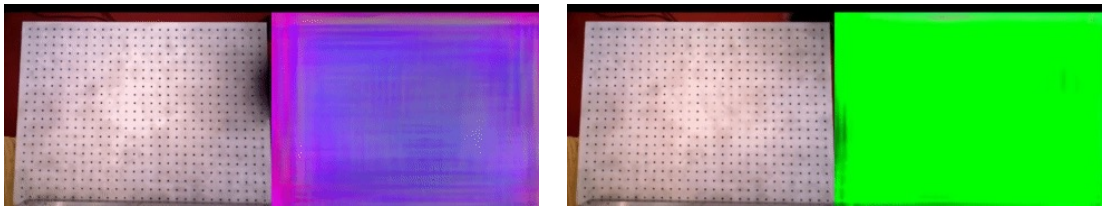
- Regions we can track easily and consistently
- Once we have the good features, we can use simple tracking methods
- Next Lecture: Optical Flow



Dense Object Nets

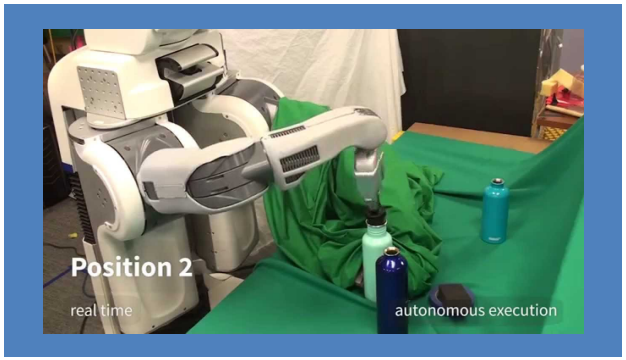
Learning *Dense* Visual Object *Descriptors*
By and *For* Robotic Manipulation. CORL 2018

Peter R. Florence, Lucas Manuelli, Russ Tedrake



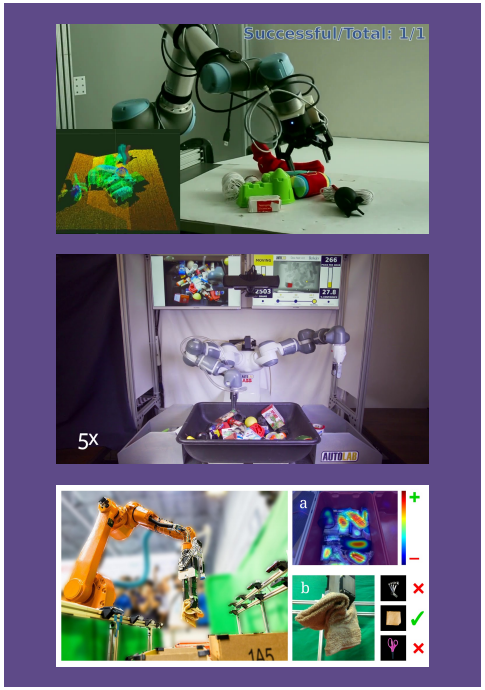
Slides adapted from CS326 by Kevin Zakka and Sriram Somasundaram

Motivation



RL

task-specific



Grasp feature-learning

no specificity

Grasp segmentation

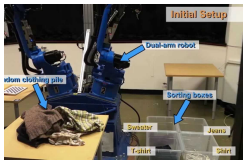
coarse

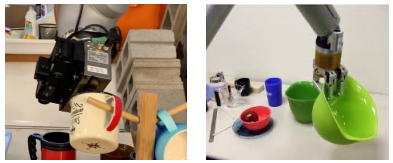
no specificity

What is the right object representation for manipulation,
and how can we scalably acquire it?

Wish List

Deformable



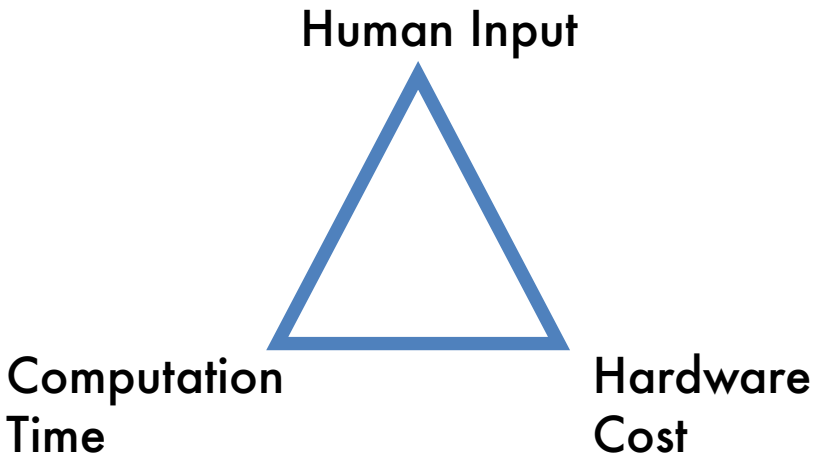


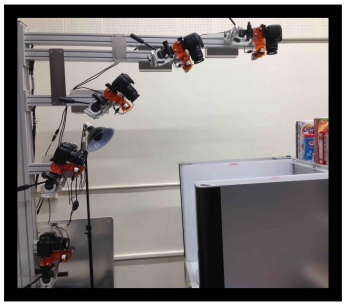
Task agnostic

Self-supervised

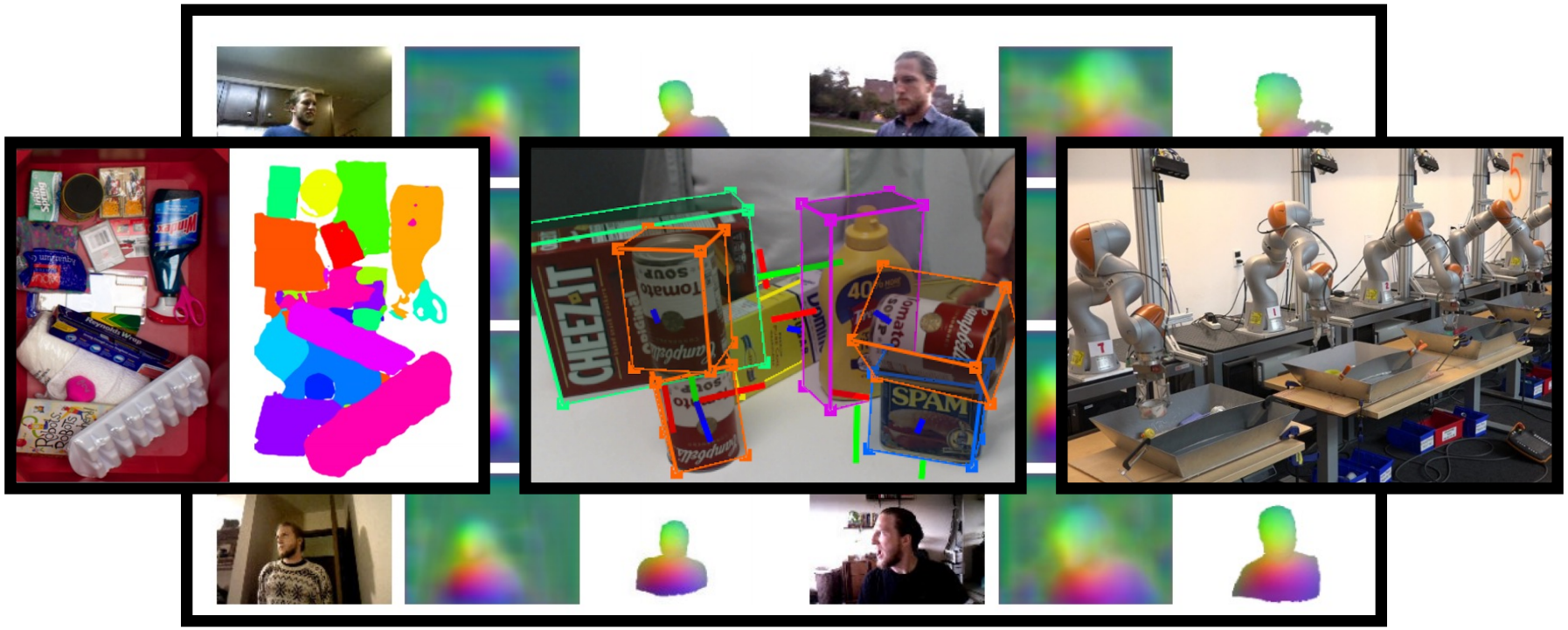
3D perception

Scalable Acquisition



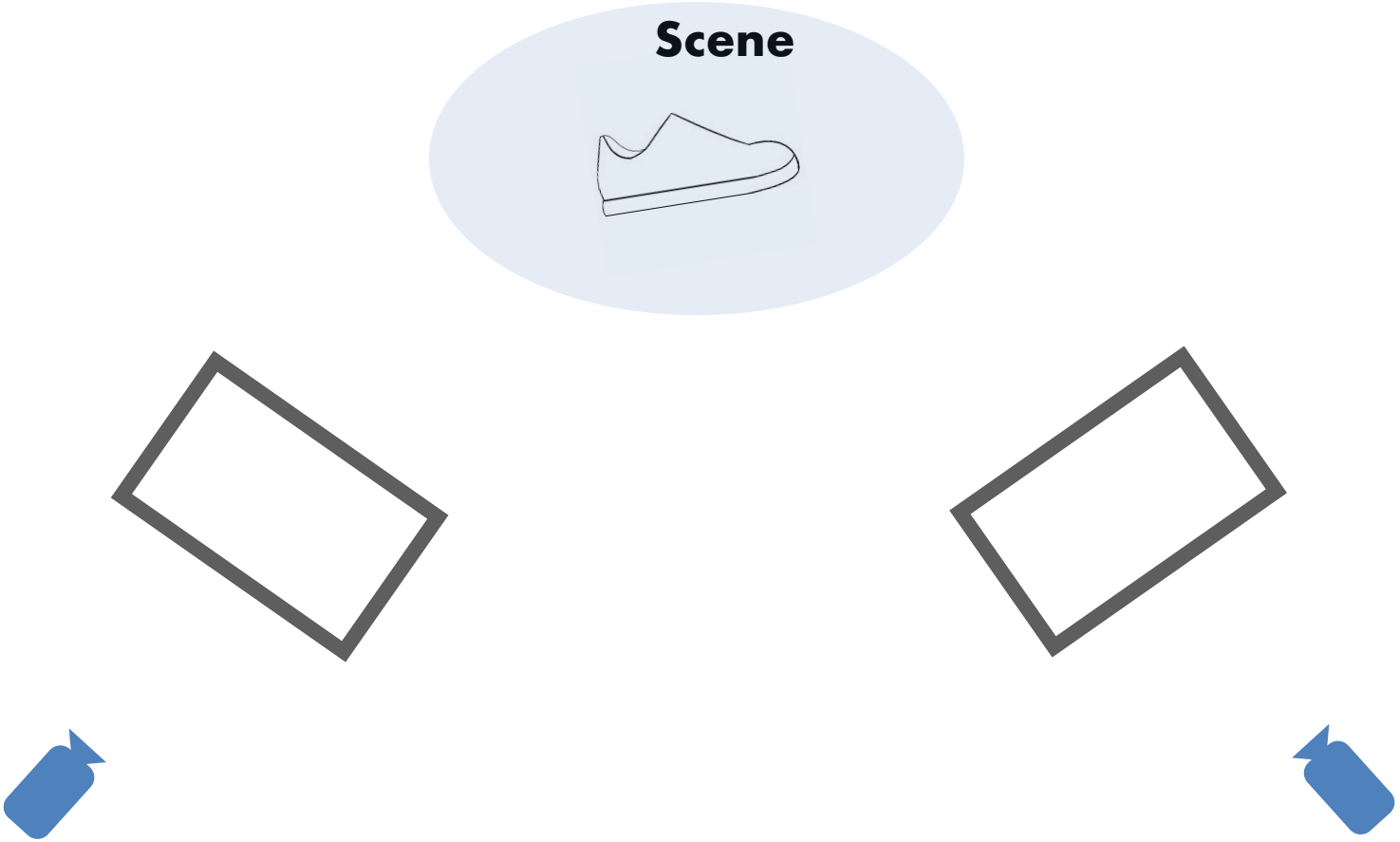


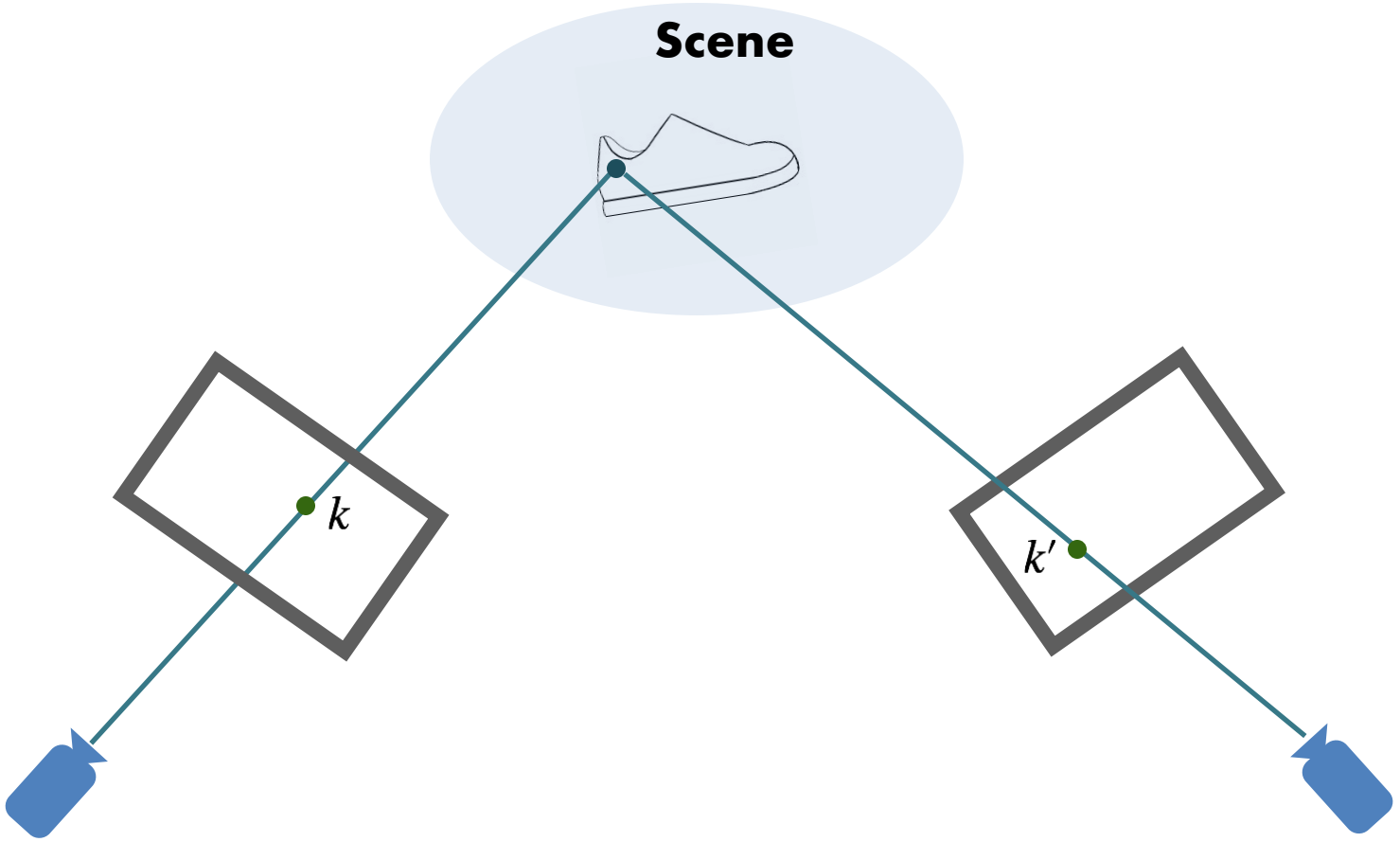
Some Representations

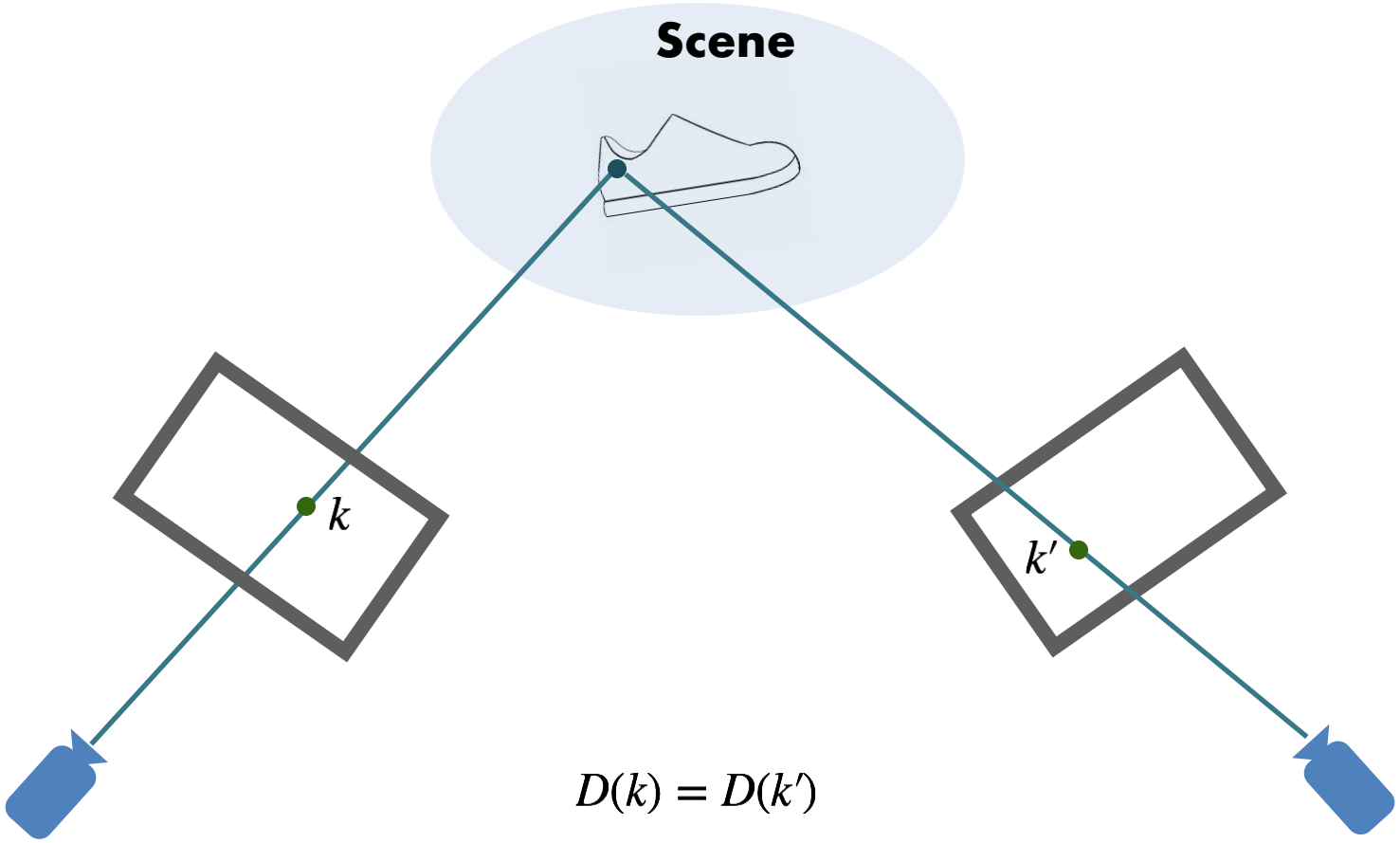


What exactly is a **descriptor**?

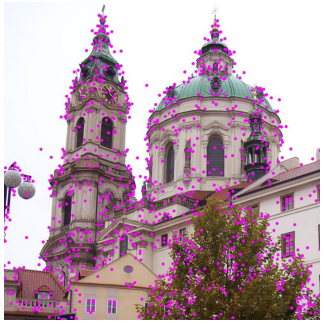
The story goes that Takeo Kanade once told a young graduate student that the three most important problems in computer vision are: "correspondence, correspondence, correspondence!" - Wang et. al. 2019

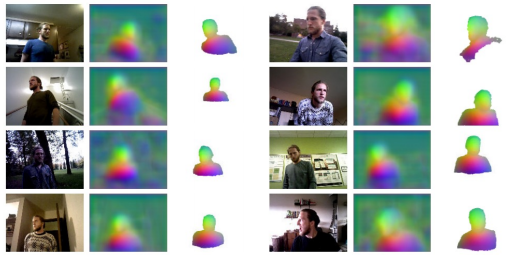

Scene


Scene
k
k′


Scene
k
k′
D(k) = D(k′)

A Brief History

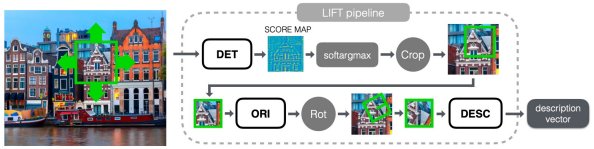




Sparse Engineered: SIFT

Dense Learned

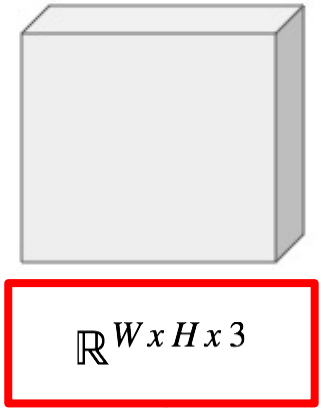
Sparse Learned: LIFT



Paper Overview

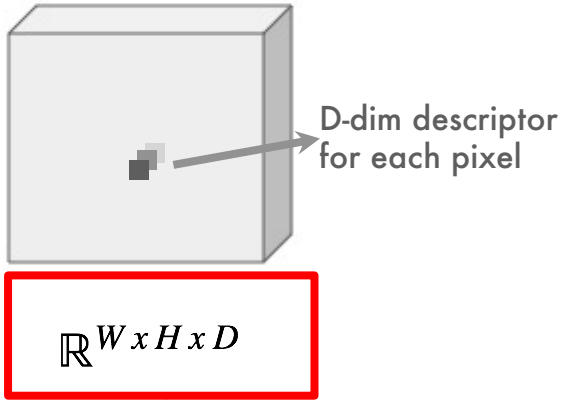
Dense Descriptors

Input is an RGB image



$f(\cdot)$

Output



Pay attention to the difference in Dimensionality

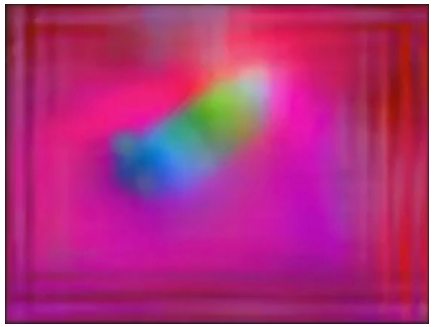
Dense Descriptors

Input is an RGB image

$\mathbb{R}^{W \times H \times 3}$

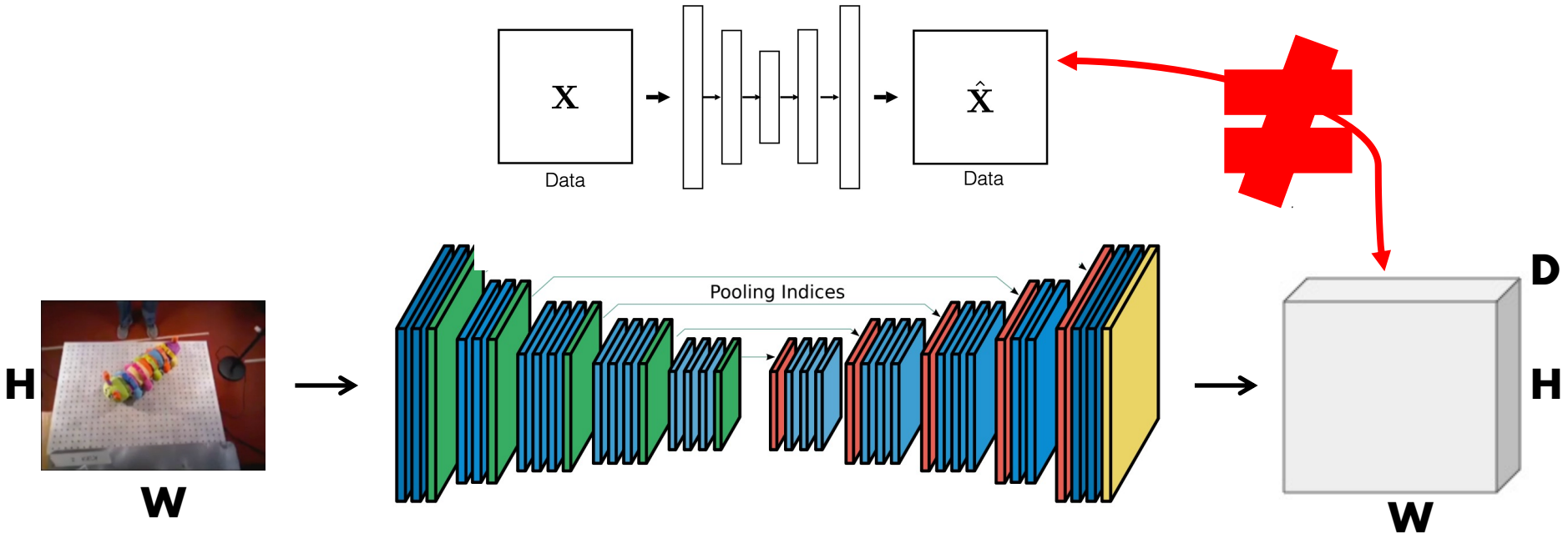
$f(\cdot)$

Output



$\mathbb{R}^{W \times H \times D}$

Network Architecture



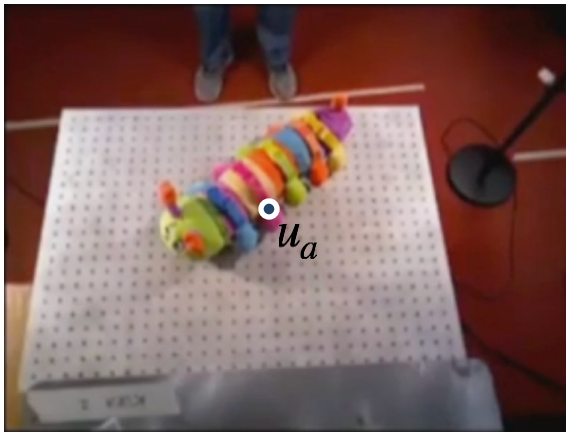
Pixelwise Contrastive Loss

I_a

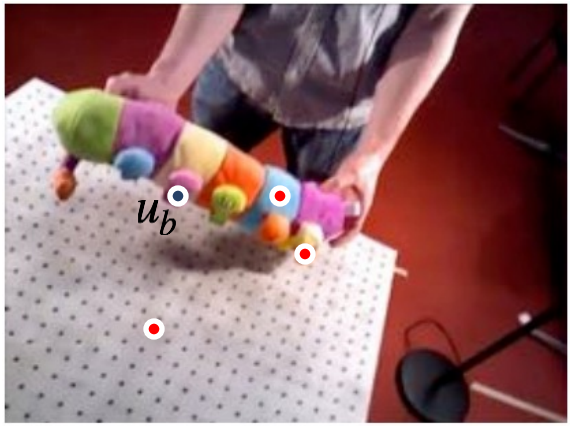
I_b

Hadsell et al., CVPR 2006

Pixelwise Contrastive Loss



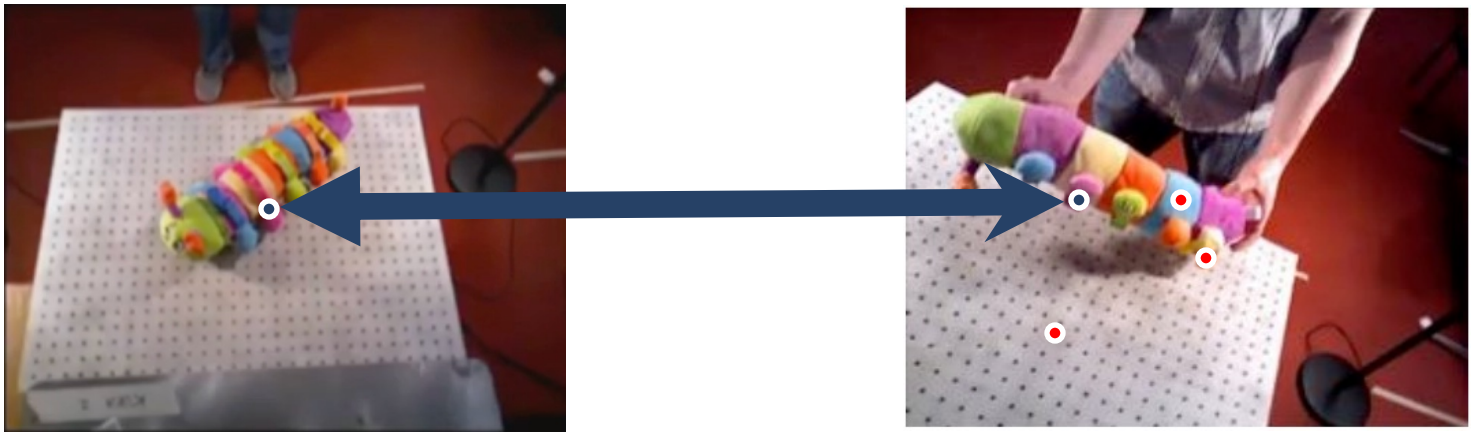
I_a



I_b

Assumption: Ground truth Correspondences Given

Pixelwise Contrastive Loss - Matches

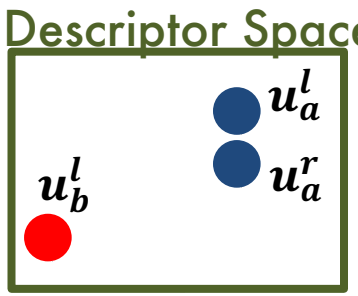


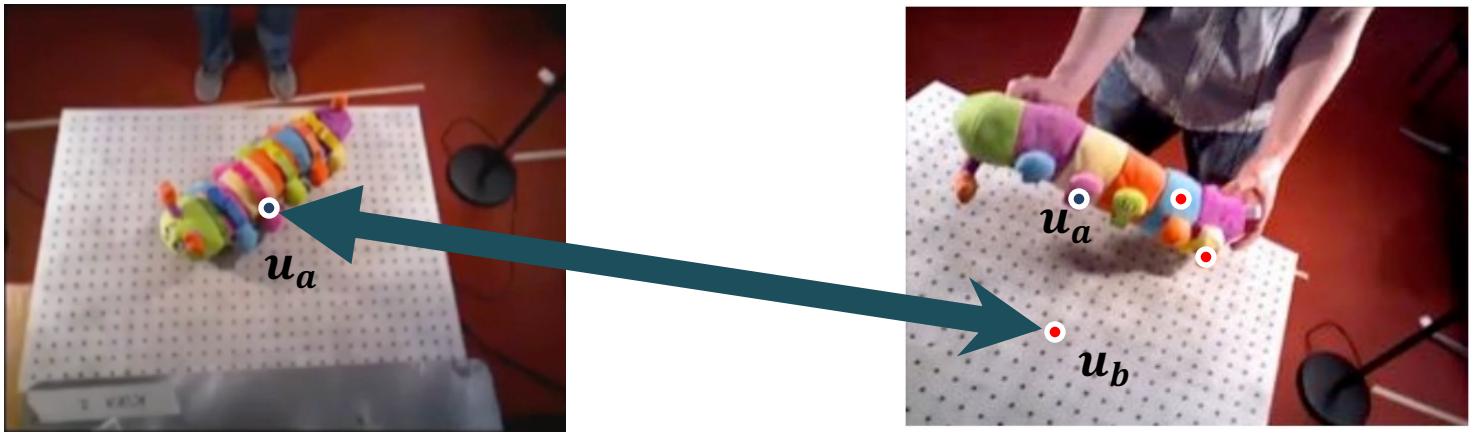
$$L_{\text{matches}}(I_a, I_b) = \frac{1}{N_{\text{matches}}} \sum_{N_{\text{matches}}} D(I_a, u_a, I_b, u_b)^2$$

Distance in Descriptor Space

Pixelwise Contrastive Loss – Non-Matches

- Training time





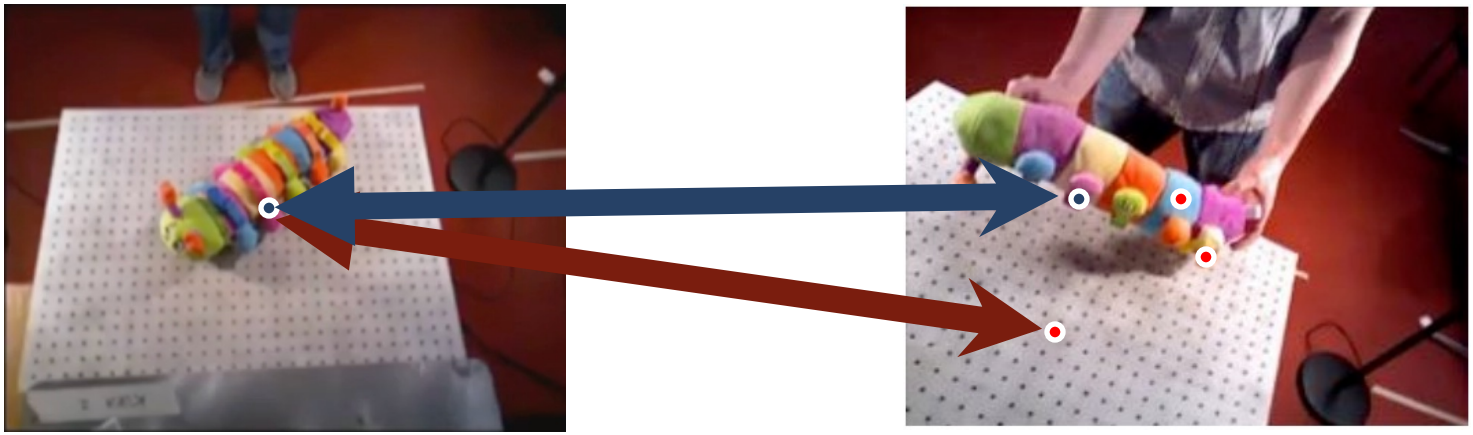
Max distance

$$L_{\text{non-matches}}(I_a, I_b) = \frac{1}{N_{\text{non-matches}}} \sum_{N_{\text{non-matches}}} \max(0, M - D(I_a, u_a, I_b, u_b)^2)$$

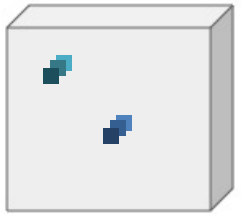
Distance in Descriptor Space

You want this to be large for non-matches

Pixelwise Contrastive Loss



$$L(I_a, I_b) = L_{\text{matches}}(I_a, I_b) + L_{\text{non-matches}}(I_a, I_b)$$

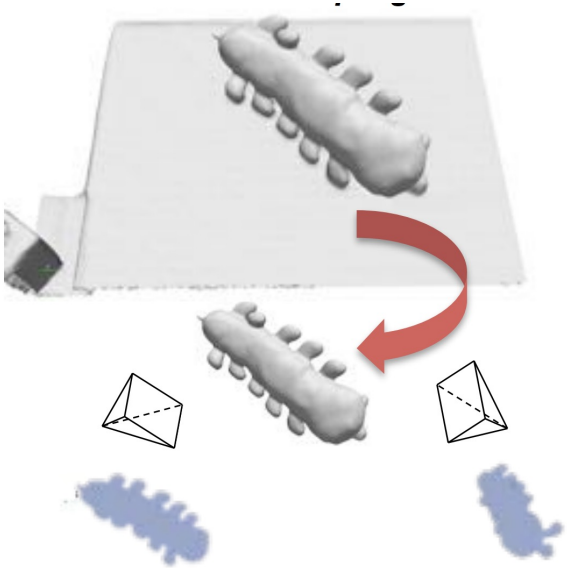


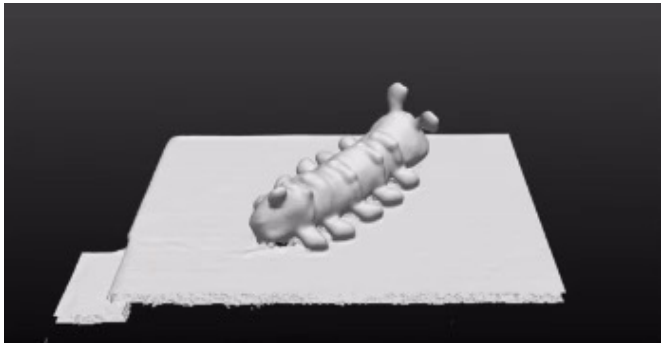
$\mathbb{R}^{W \times H \times D}$

Loss: Reconstruction + Contrastive Loss

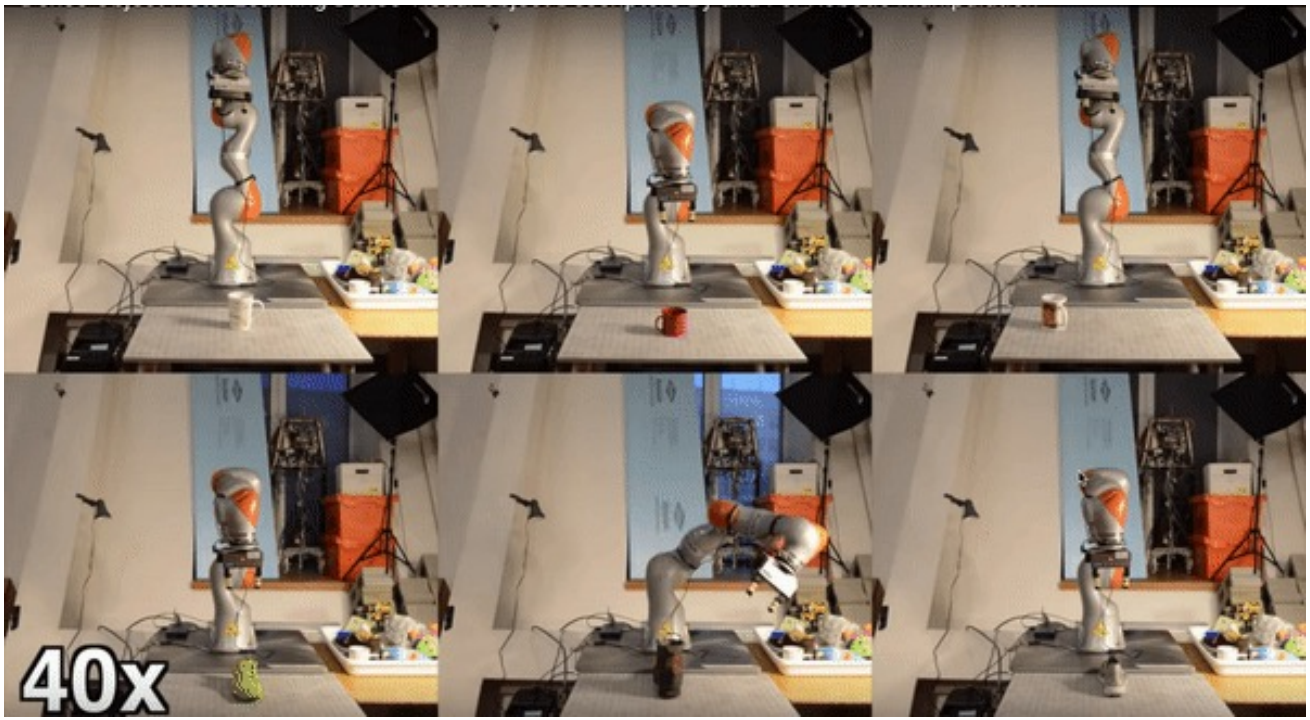
How can we generate these ground-truth correspondences?

3D Reconstruction based Change Detection and Masked Sampling

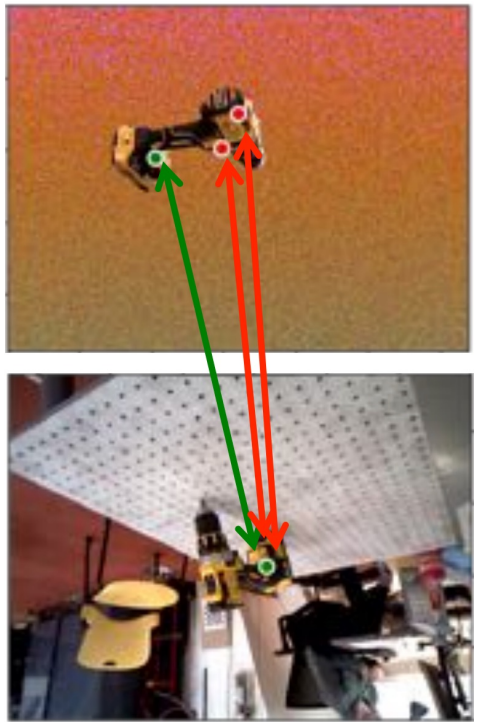




Autonomous and Self-supervised Data Collection



Background Randomization



Further Training Techniques

- Hard-Negative Scaling

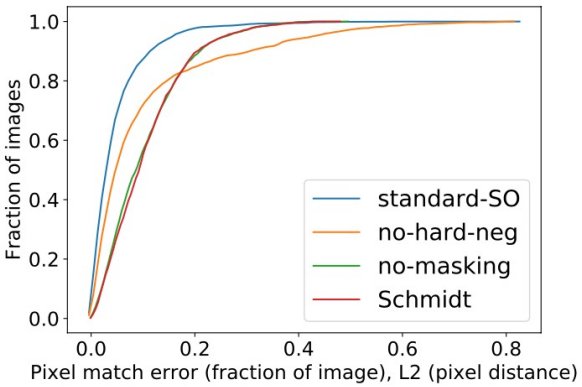
$$L_{\text{non-matches}}(I_a, I_b) = \frac{1}{N_{\text{hard-negatives}}} \sum_{N_{\text{non-matches}}} \max(0, M - D(I_a, u_a, I_b, u_b)^2)$$

- Data Augmentation

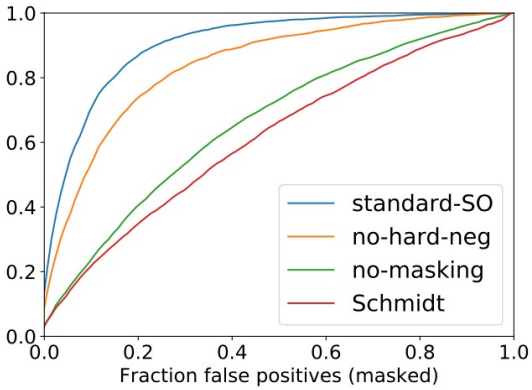
Results

	single or multi object dataset	masked match sampling	scale by hard negatives	cross-object loss
standard-SO	single	✓	✓	
no-masking	single		✓	
no-hard-neg	single	✓		
Schmidt	single			
consistent	multi	✓	✓	
specific	multi	✓	✓	✓

(a)



(b)



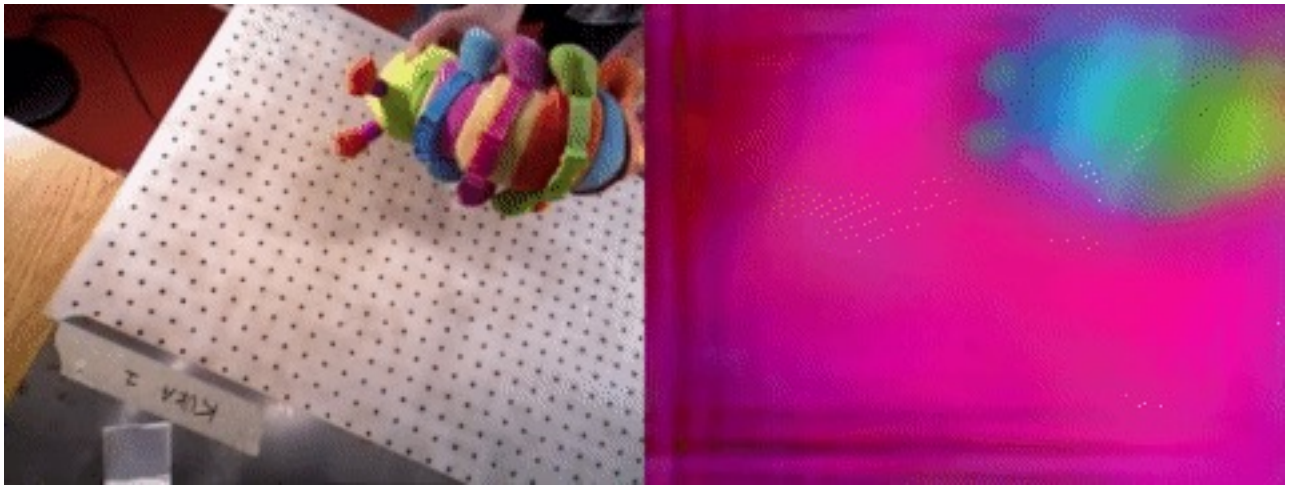
(c)

Experiments

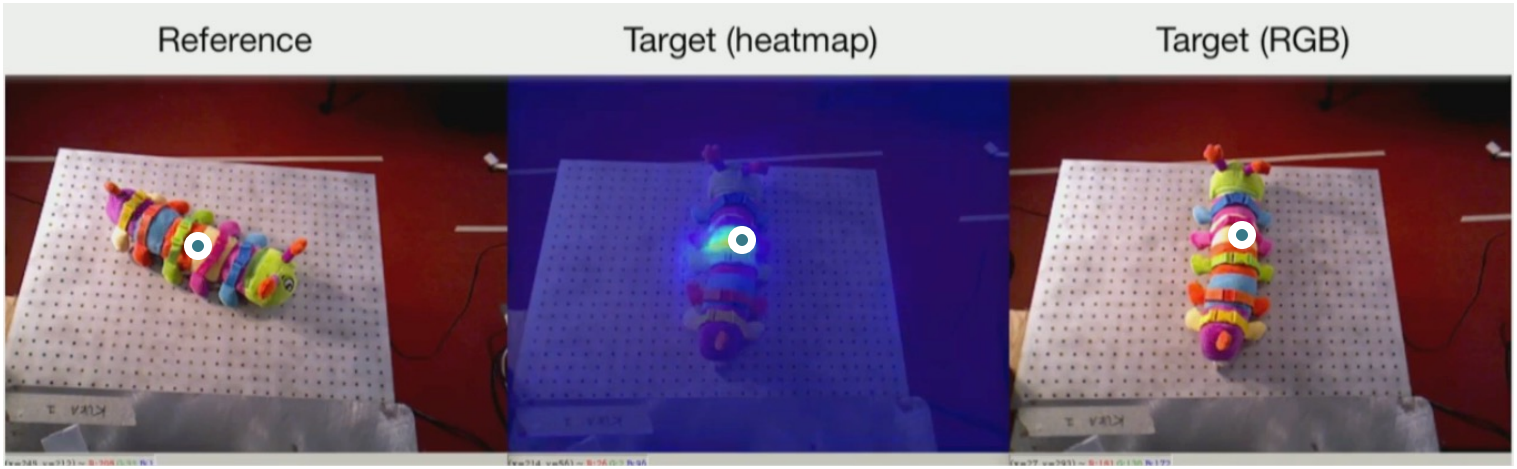
1) Can we acquire descriptors that can generalize across classes of objects and can be distinct for each object instance?

2) Can we apply dense descriptors to robotic manipulation?

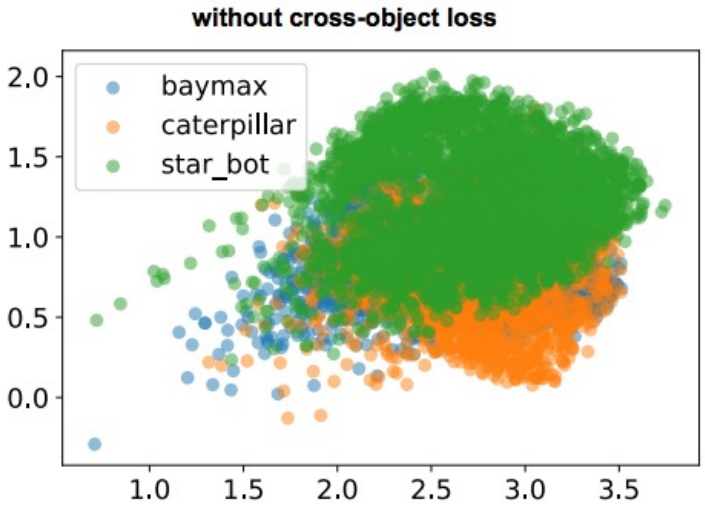
Single Object

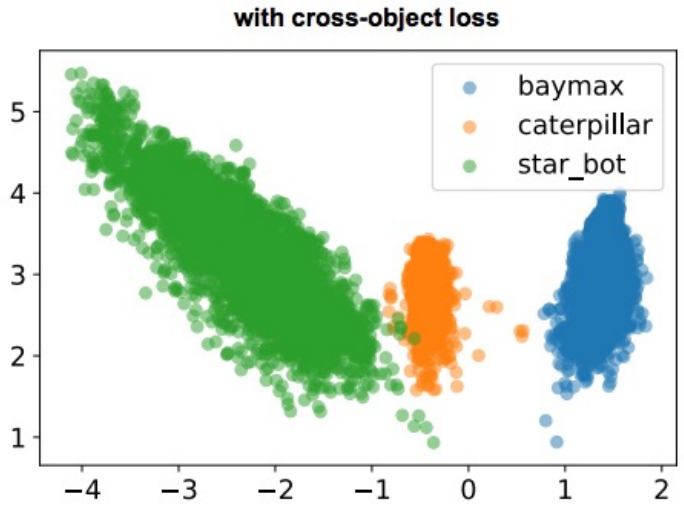


Learned Dense Correspondences

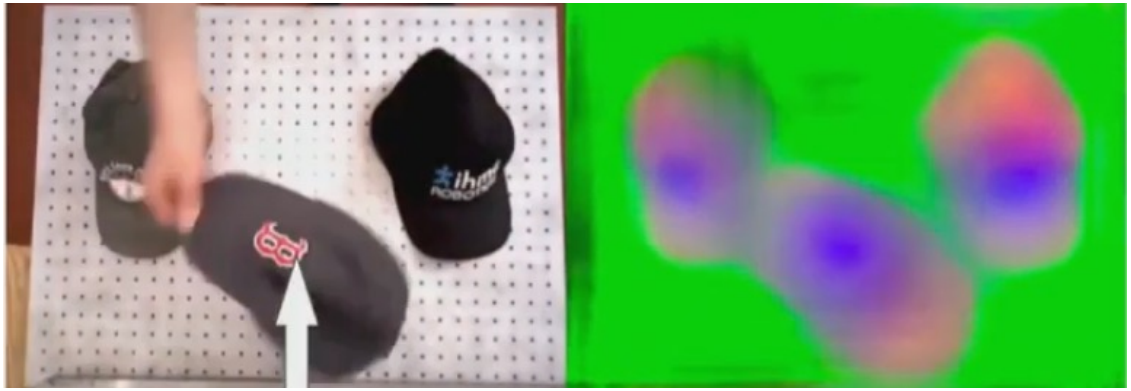


Multi-Object Unique Descriptors





Class consistent descriptors



Experiments

1) Can we acquire descriptors that can generalize across classes of objects and can be distinct for each object instance?

Yes

2) Can we apply dense descriptors to robotic manipulation?

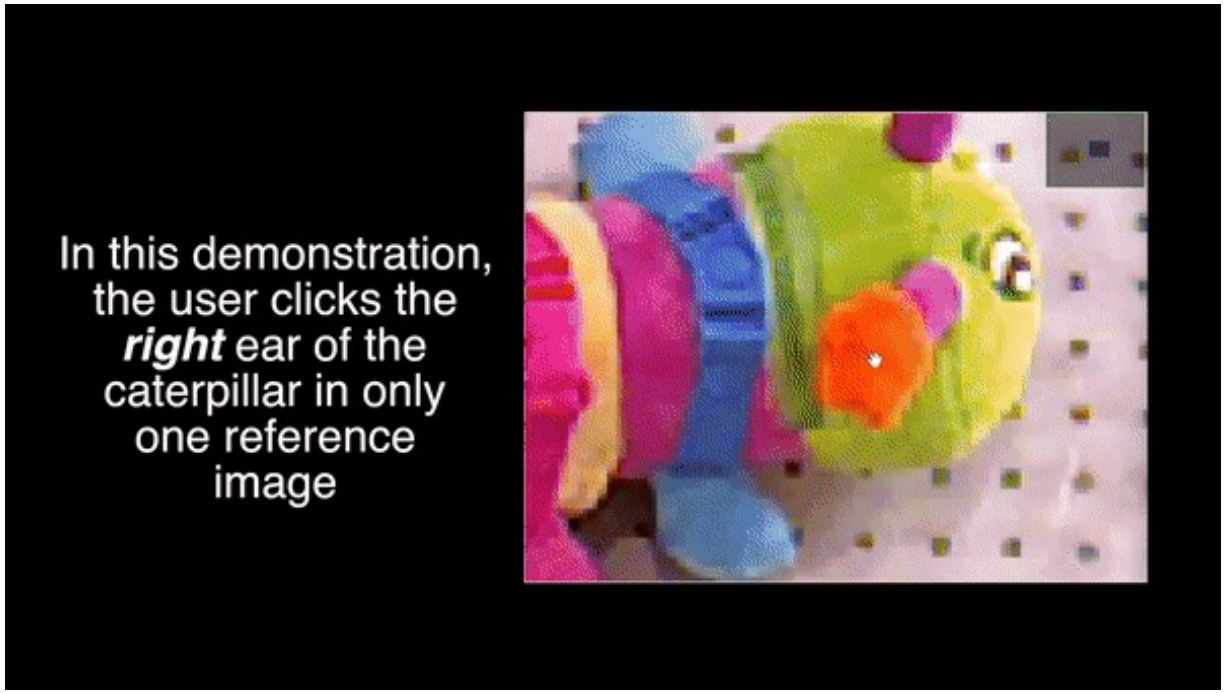
Robotic Manipulation

Results: Robotics Showcase

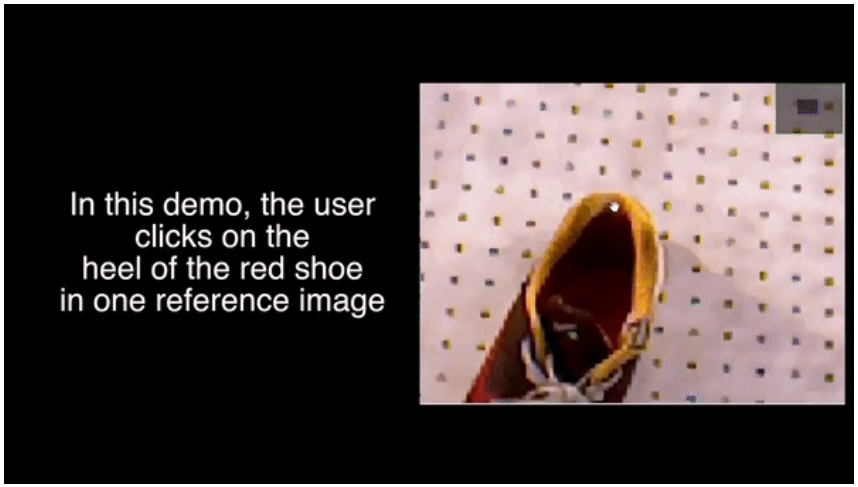
Goal: Perform grasps on target object with a real robot based on selected grasp points on a reference object

- Dense correspondence for manipulation
- Multi object dense descriptor manipulation
- Class consistent descriptor manipulation

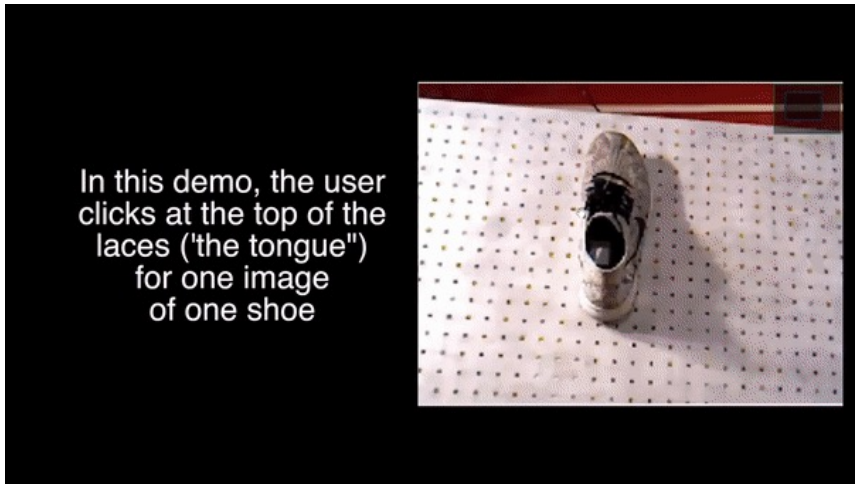
Dense Correspondence for Manipulation

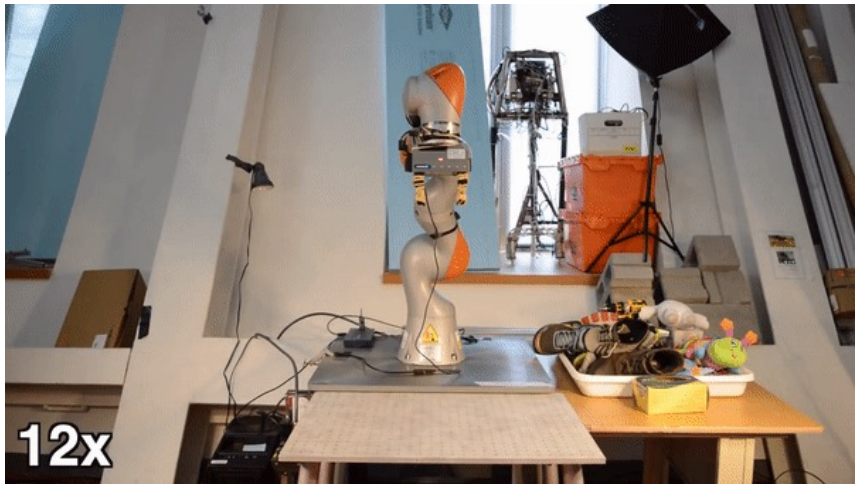


Instance Specific Manipulation



Class Consistent Manipulation





Experiments

1) Can we acquire descriptors that can generalize across classes of objects and can be distinct for each object instance?

Yes

2) Can we apply dense descriptors to robotic manipulation?

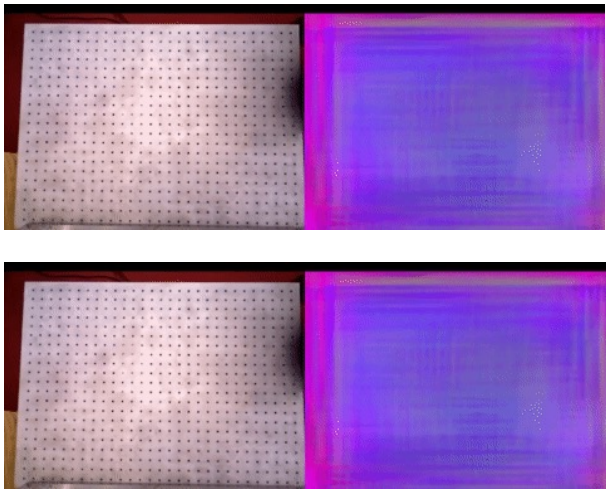
Yes, we can use descriptors to know where to grasp

Limitations

- Learned correspondences not always **unimodal**
- Training is finicky: **sensitive** to scale of match and non-matches
- Don't always have consistency **guarantee** (e.g. anthropomorphic toys)

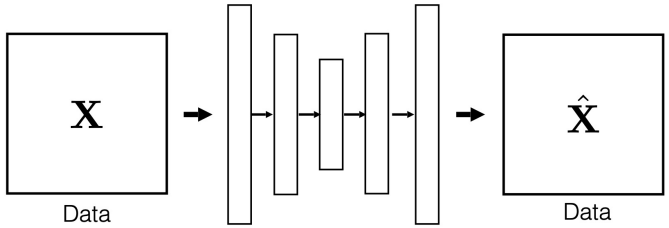
Let’s use representation learning!

Depth Estimation



Feature Tracking

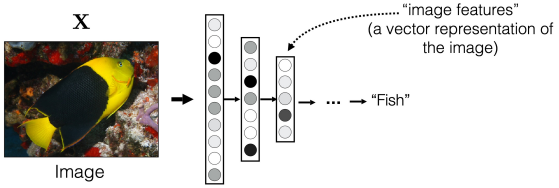
Summary



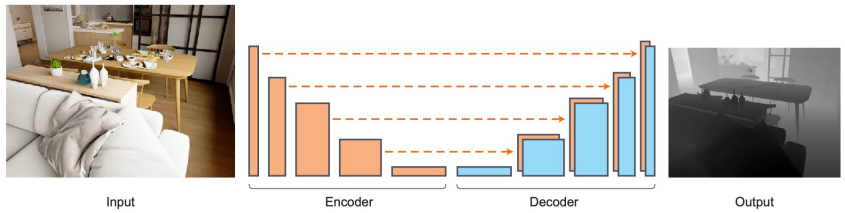
- Hourglass structure
- Mapping either to depth or to descriptors
  - Descriptors are used for localization, robot grasping, tracking
  - Keypoint matching
- Training is unsupervised, self-supervised by exploiting structure that you know about the problem
  - Eases demand for ground truth labels

Learning Goals for Upcoming Lectures

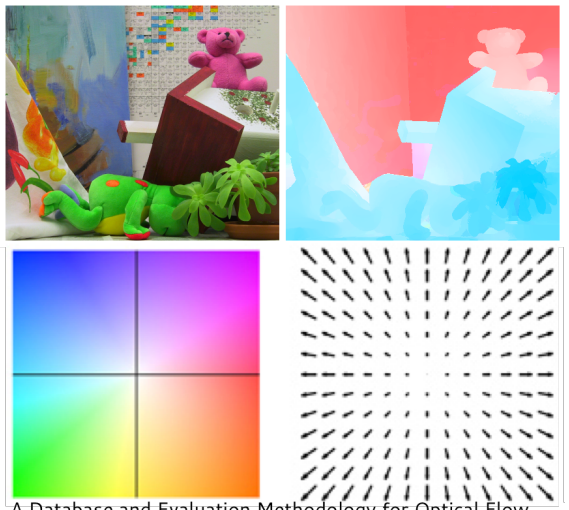
Representations & Representation Learning



Monocular Depth Estimation, Feature Tracking

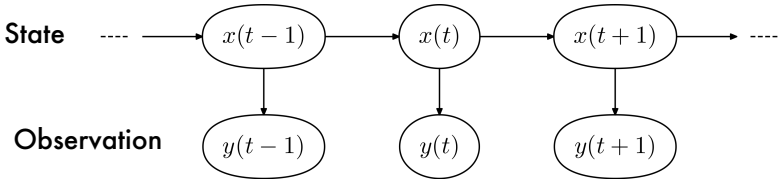


Optical & Scene Flow

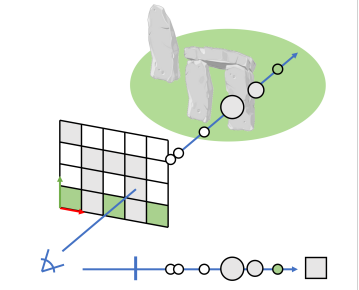


A Database and Evaluation Methodology for Optical Flow.
Baker et al. IJCV. 2011

Optimal Estimation

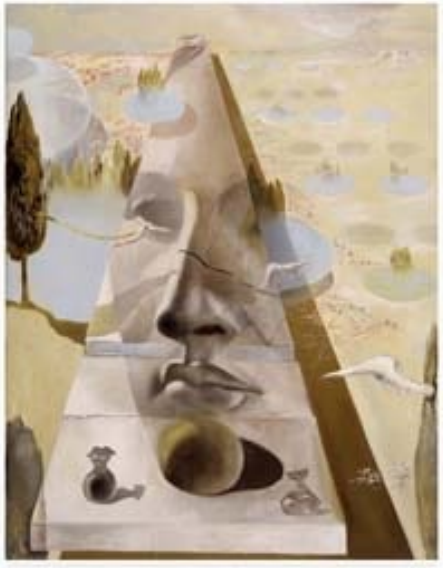


Neural Radiance Fields



CS231

Introduction to Computer Vision



Next lecture:

Optical Flow and Scene Flow