CS231A

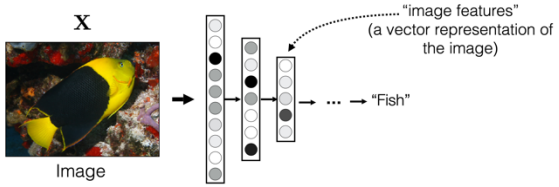
Computer Vision:

From 3D Reconstruction to Recognition

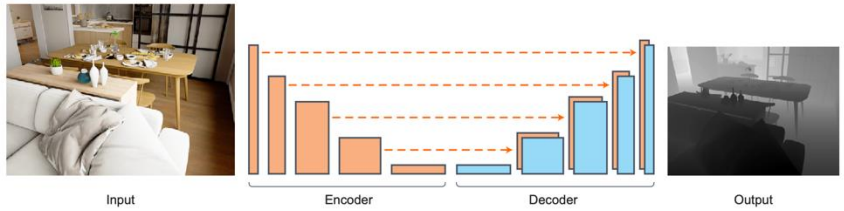
Representation Learning for Finding Correspondences and Depth Estimation

Learning Goals for Upcoming Lectures

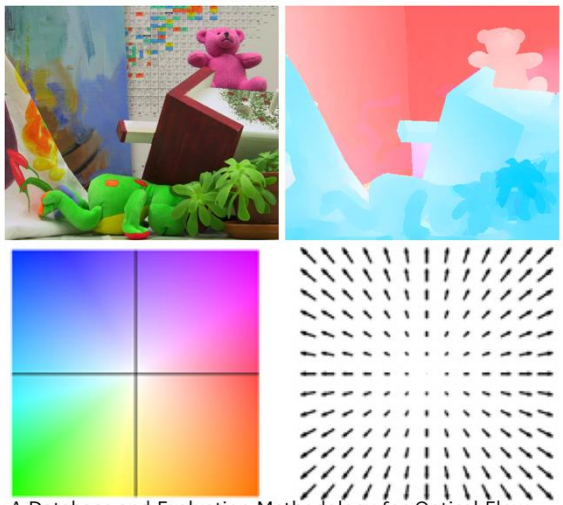
Representations & Representation Learning



Using Representation Learning for Depth Estimation and Finding Correspondences

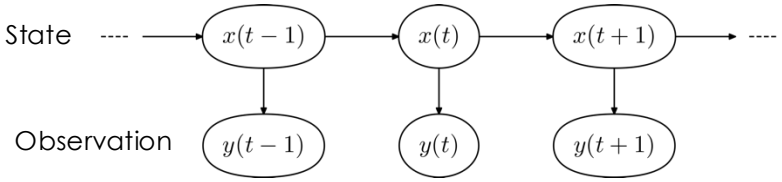


Optical & Scene Flow

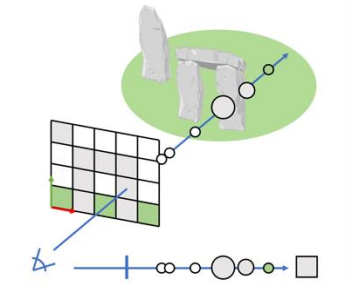


A Database and Evaluation Methodology for Optical Flow.
Baker et al. IJCV. 2011

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?
- How can we learn good representations form data?

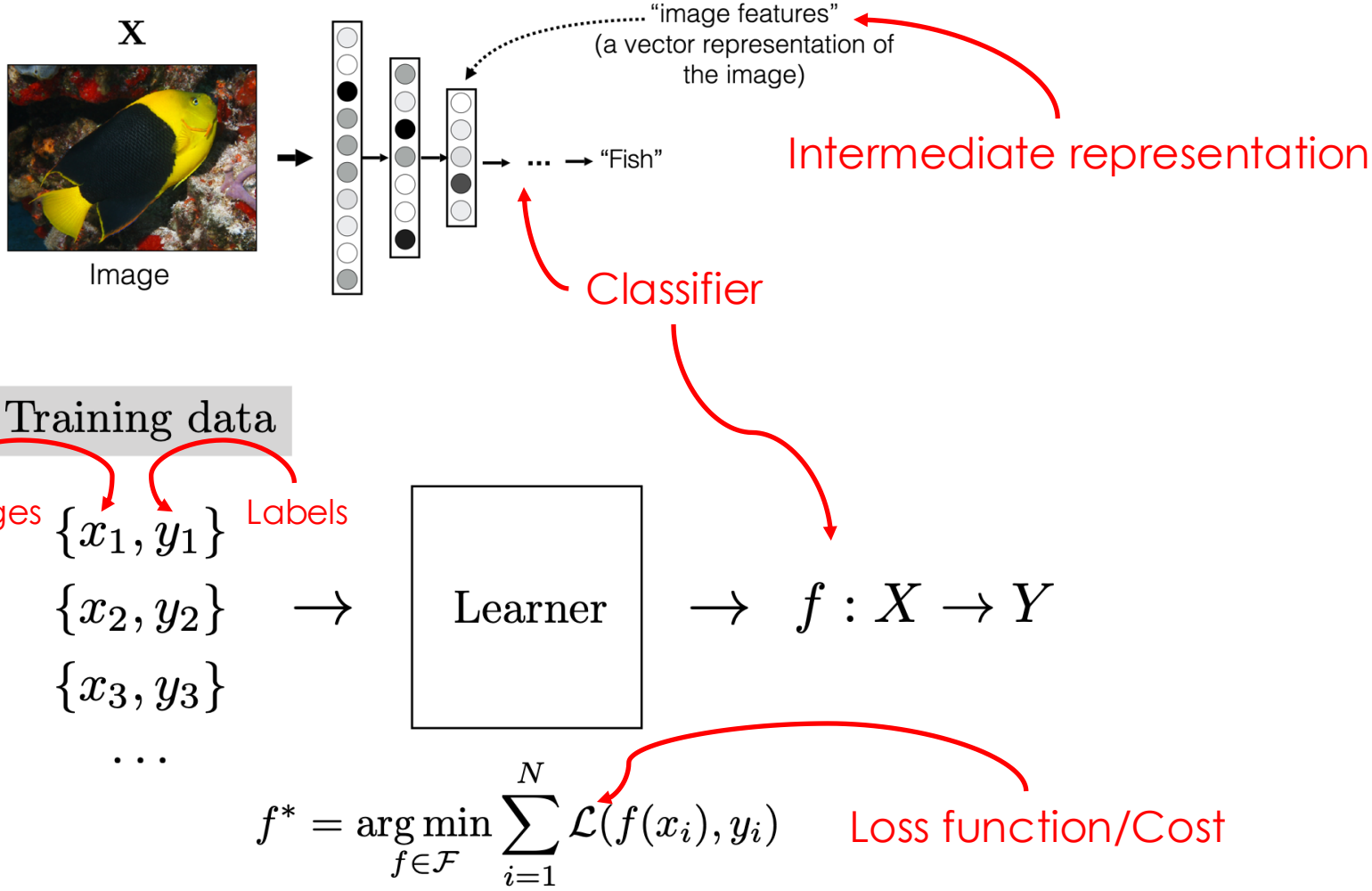
Summary of what you learned

- **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

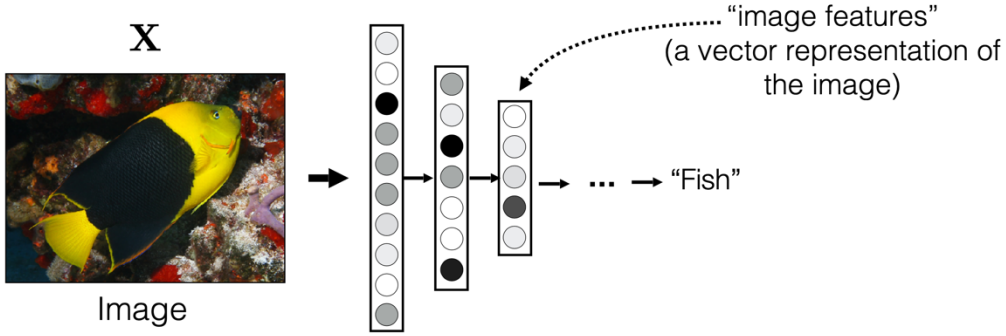
Summary of what you learned

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

Supervised learning of a representation



Learning without Labels



Training data

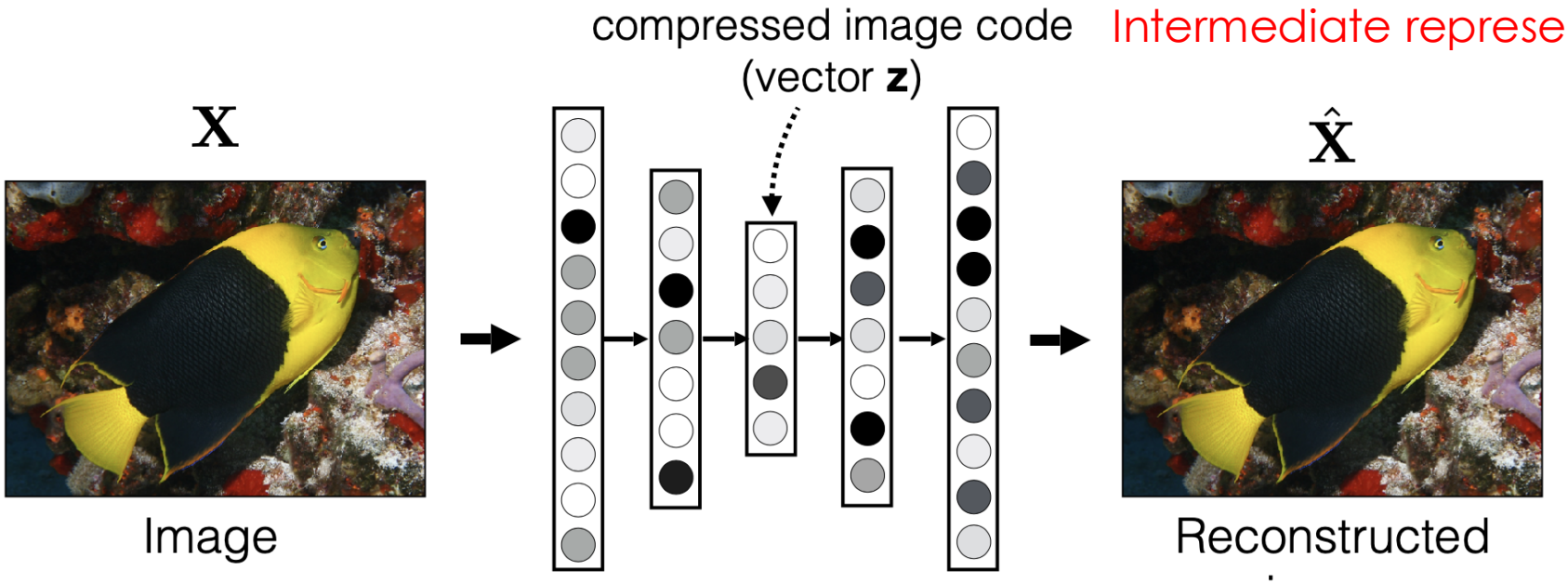
$$\{x_1, y_1\}$$
$$\{x_2, y_2\}$$
$$\{x_3, y_3\}$$
$$\dots$$

$$\rightarrow \quad \rightarrow \quad f : X \rightarrow Y$$

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

Unsupervised Representation Learning

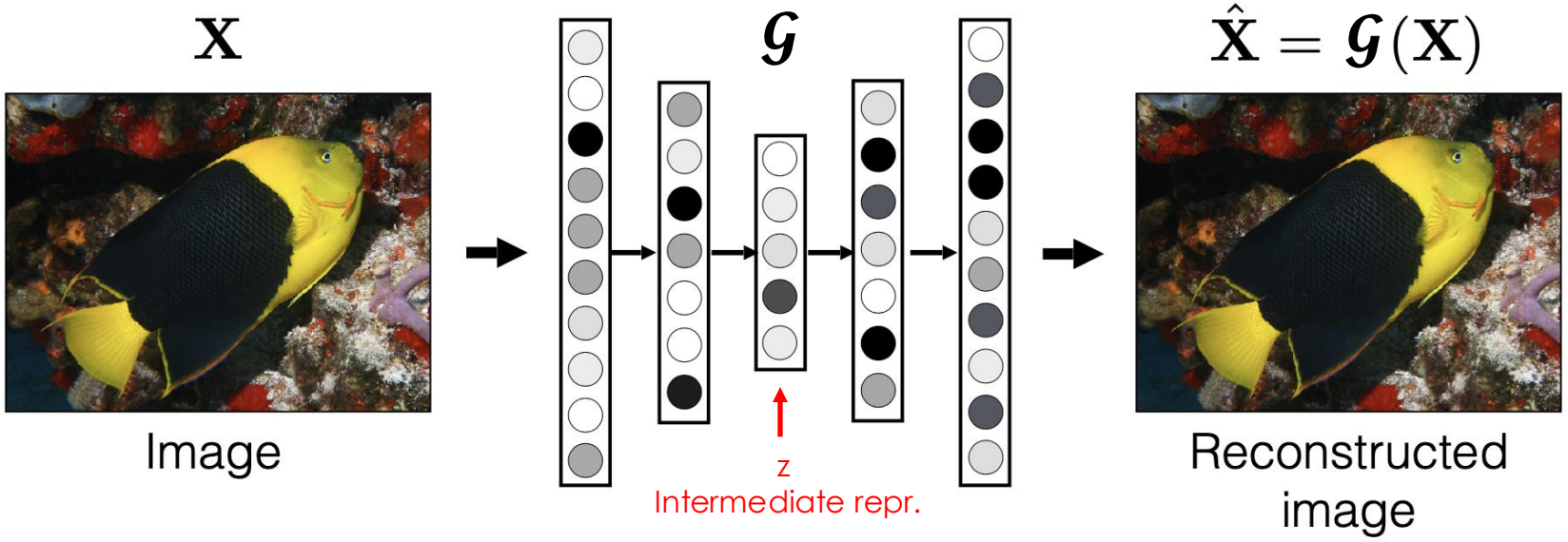
No category or symbolic label. Instead: learn to reconstruct.



One kind of unsupervised model: "Autoencoder"

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

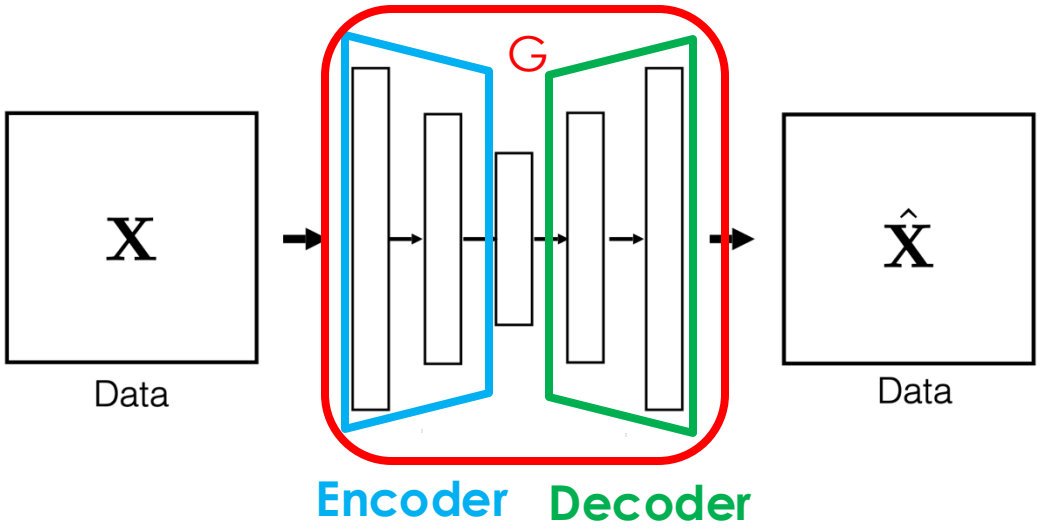
Autoencoder

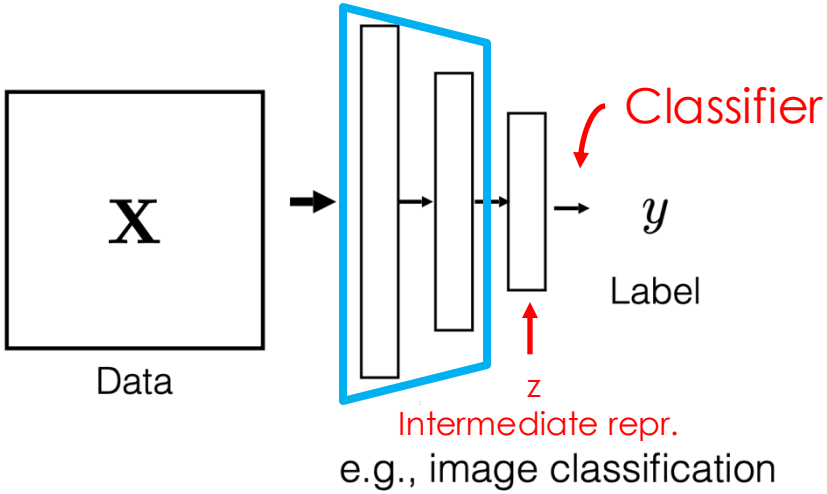


$$\arg\min_{\mathcal{G}} \mathbb{E}_{\mathbf{X}}[||\mathcal{G}(\mathbf{X}) - \mathbf{X}||]$$

Reconstruction loss to minimize by finding optimal G

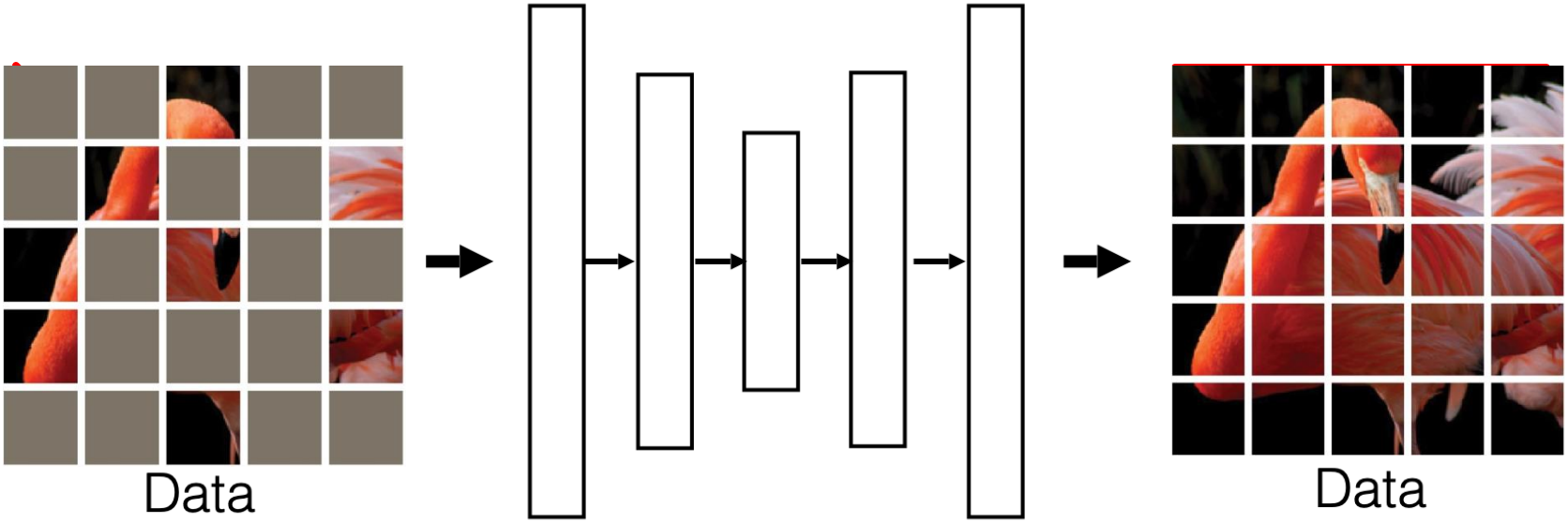
Data Compression & Task Transfer





e.g., image classification

Self-Supervision



$$G(X) = \hat{X}$$

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

Supervised, Unsupervised and Self-Supervised Representation Learning

0 surveys completed

0 surveys underway

Which of the options are supervised learning objectives in representation learning?

- Classification loss
- Image Reconstruction Loss
- Object Detection loss
- Depth Estimation Error from Stereo images compared to g…
- Image Synthesis Error for right image of a stereo pair give…
- Image synthesis Error of future images from current image
- Semantic Segmentation Loss

Which of the options are unsupervised learning objectives in representation learning?

- Classification loss
- Image Reconstruction Loss
- Object Detection loss
- Depth Estimation Error from Stereo images compared to g…
- Image Synthesis Error for right image of a stereo pair give…
- Image synthesis Error of future images from current image
- Semantic Segmentation Loss

Which of the options are selfsupervised learning objectives in representation learning?

- Classification loss
- Image Reconstruction Loss
- Object Detection loss
- Depth Estimation Error from Stereo images compared to g…
- Image Synthesis Error for right image of a stereo pair give…
- Image synthesis Error of future images from current image
- Semantic Segmentation Loss

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

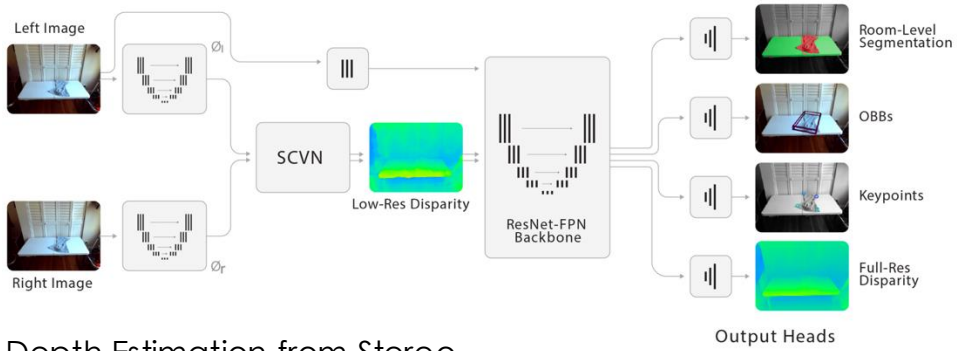
Unsupervised / Self-Supervised Learning (Cake)

Predicting parts of observed input or predicting future observations or events

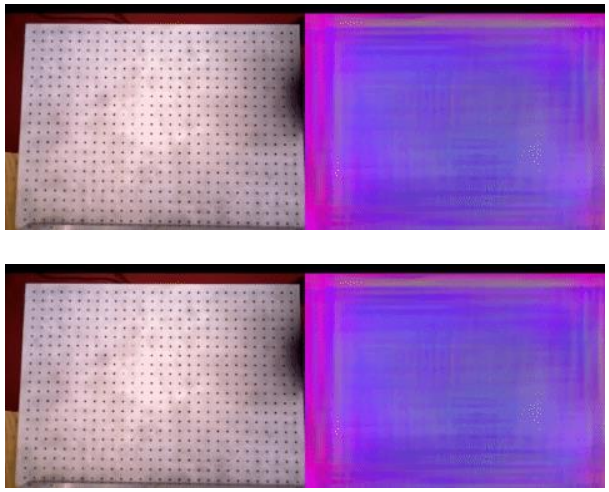
Millions of bits per sample

Visualisation Idea by Yann LeCun
Photo by Kristina Paukshtite from Pexels

Let's use representation learning!



Depth Estimation from Stereo
Supervised Learning



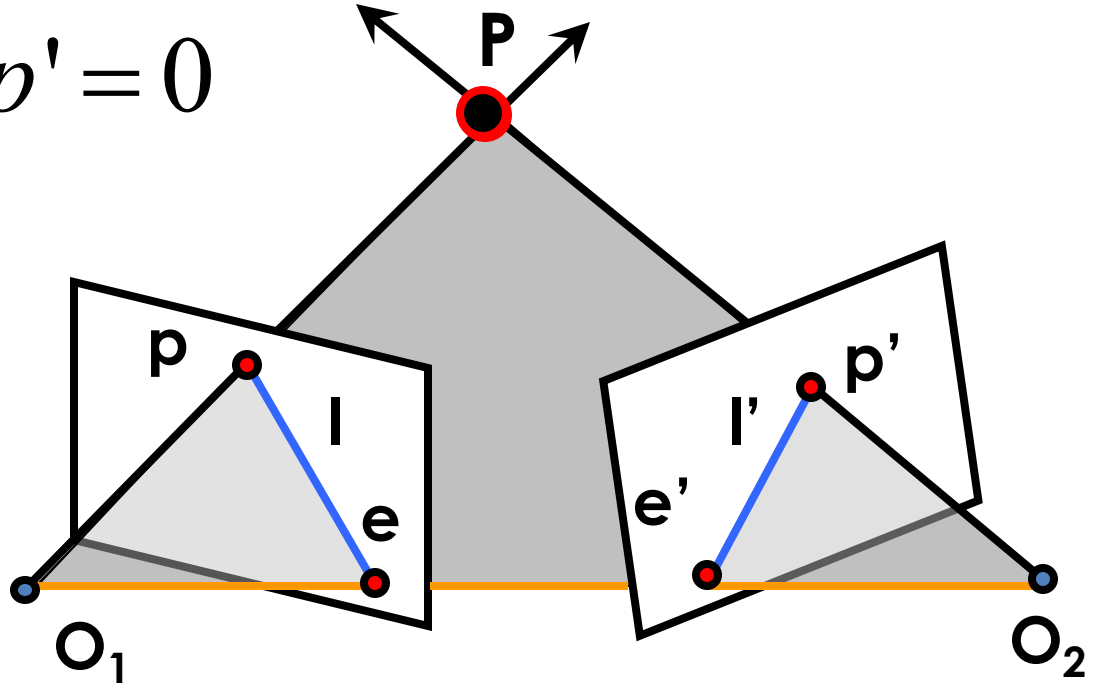
Monocular Depth Estimation
Unsupervised Learning

Image by Yunuk Cha.

Finding Correspondences across Frames
Self-Supervised Learning

Epipolar Constraint (Lecture 6)

$$p^T \square F\, p' = 0$$

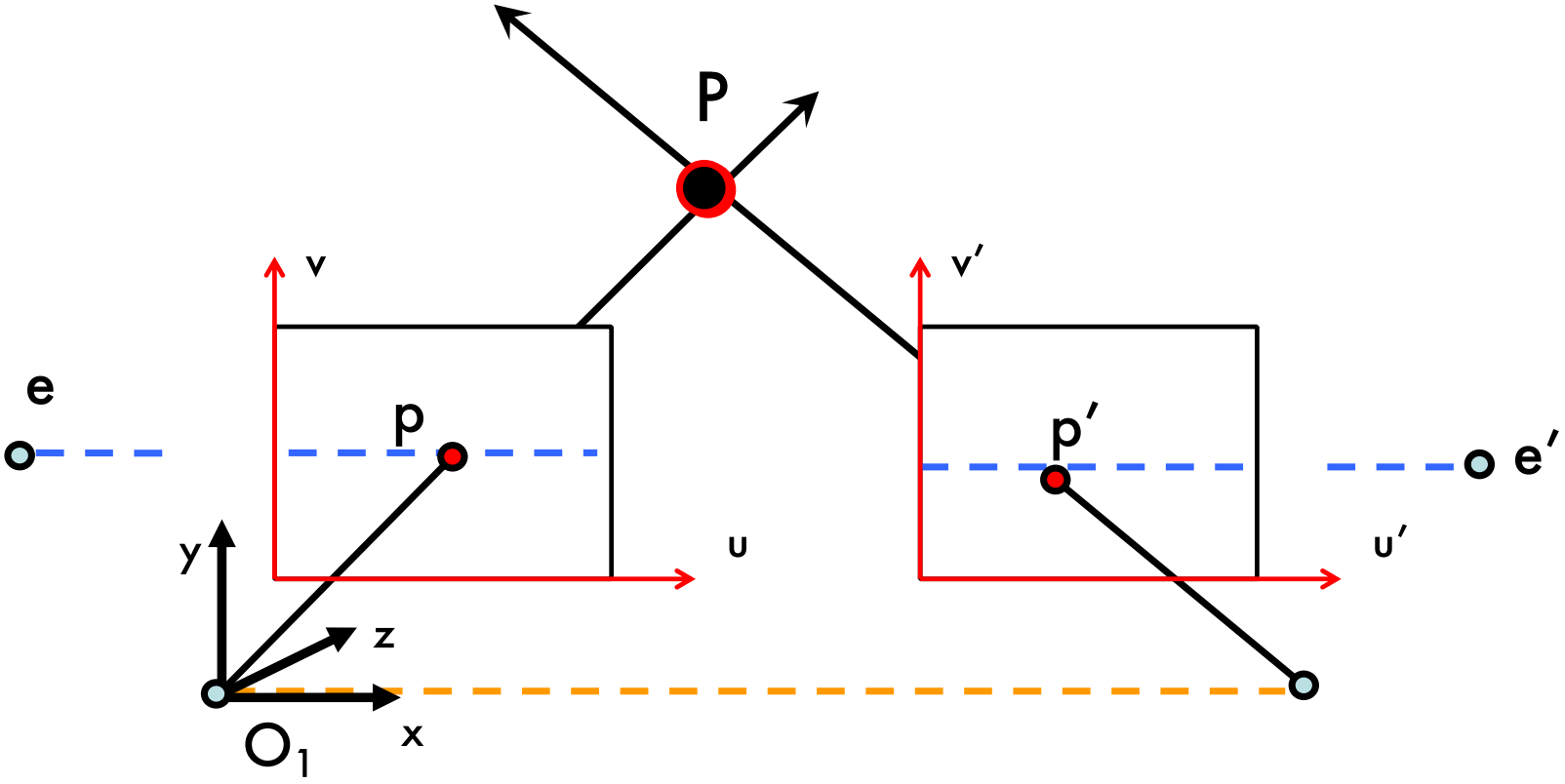


- l = F p' is the epipolar line associated with p'
- l'= F^T p is the epipolar line associated with p
- F e' = 0 and F^T e = 0
- F is 3x3 matrix; 7 DOF
- F is singular (rank two)

Rectification: making two images “parallel”

Courtesy figure S. Lazebnik

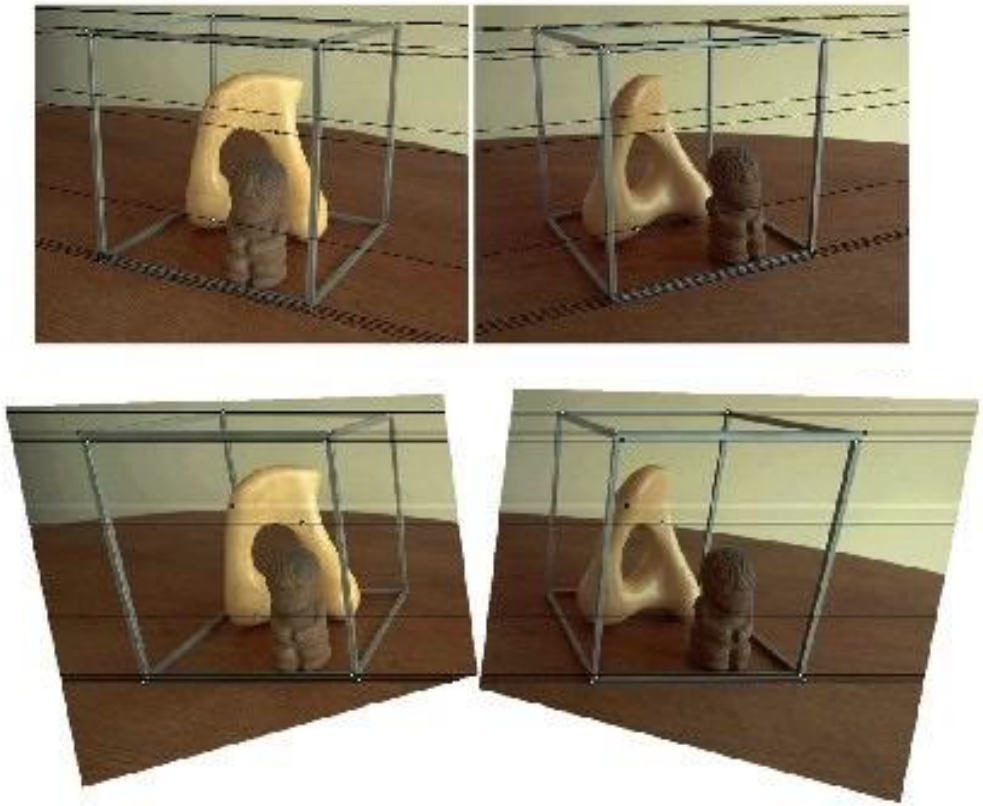
Parallel image planes



Rectification: making two images "parallel"

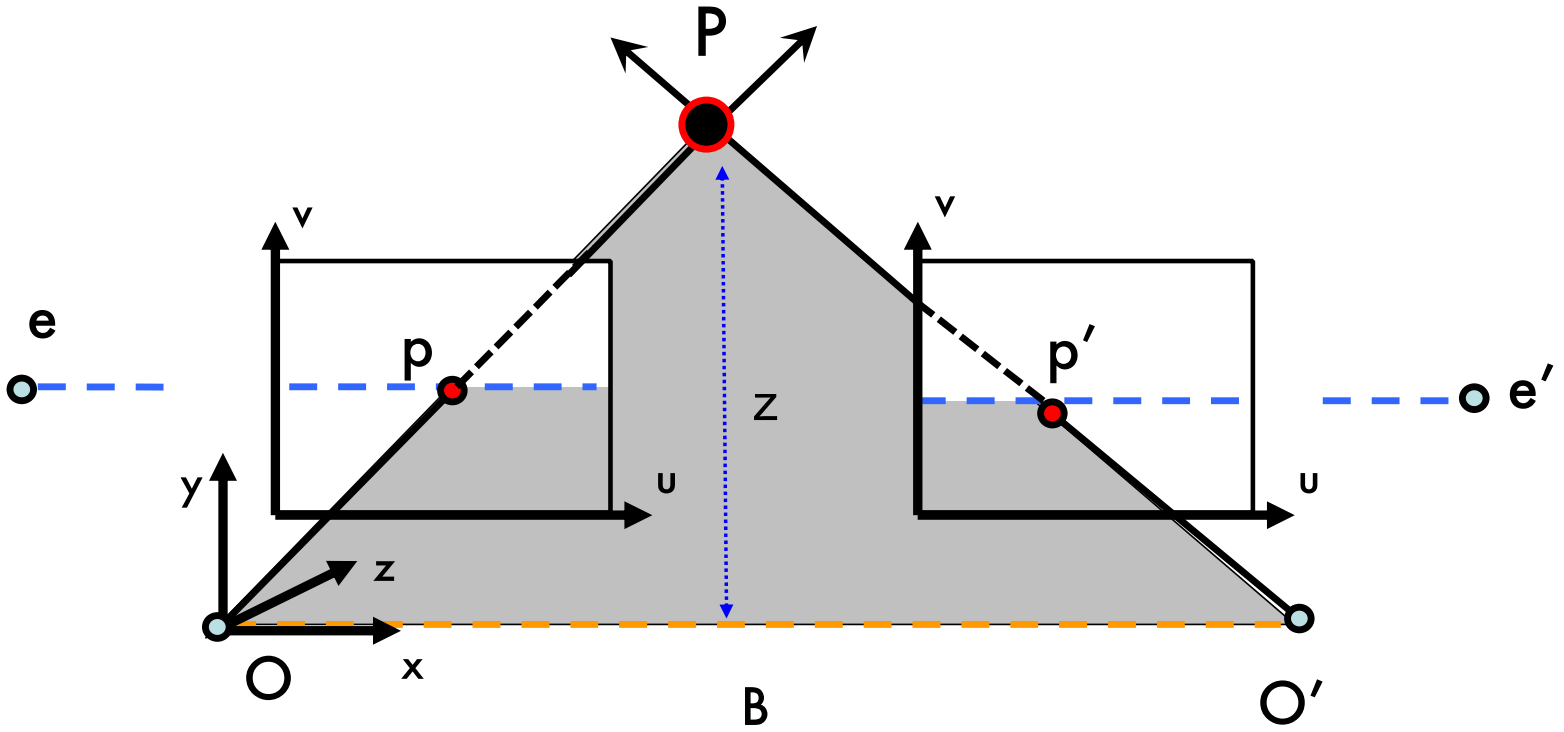
- Epipolar constraint → v = v′

Why are parallel images useful?



- Makes triangulation easy
- Makes the correspondence problem easier

Point triangulation



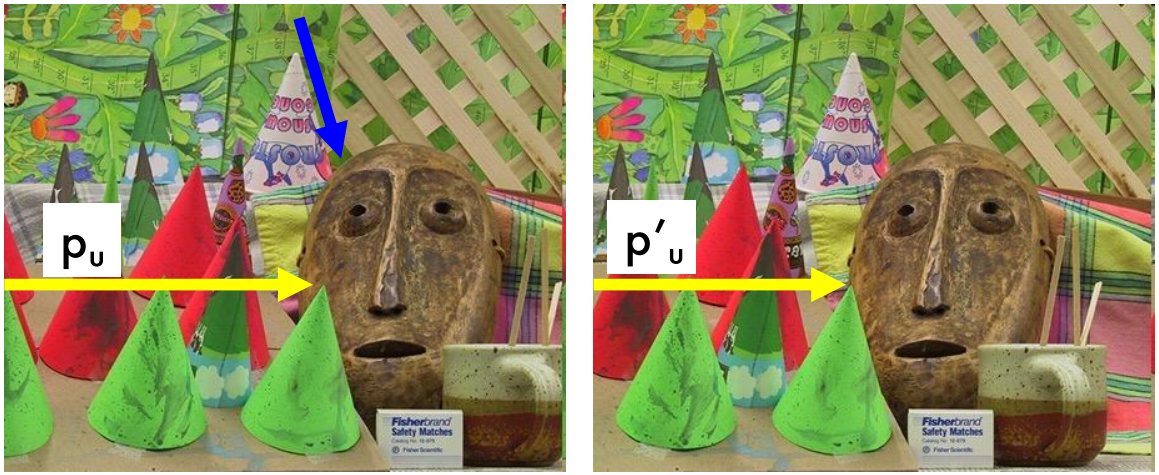
$$p = \begin{bmatrix} p_u \\ p_v \\ 1 \end{bmatrix} \quad p' = \begin{bmatrix} p'_u \\ p_v \\ 1 \end{bmatrix}$$

$$\text{disparity} = p_u - p'_u \propto \frac{B \cdot f}{z} \qquad \text{[Eq. 1]}$$

Disparity is inversely proportional to depth z!

Disparity maps

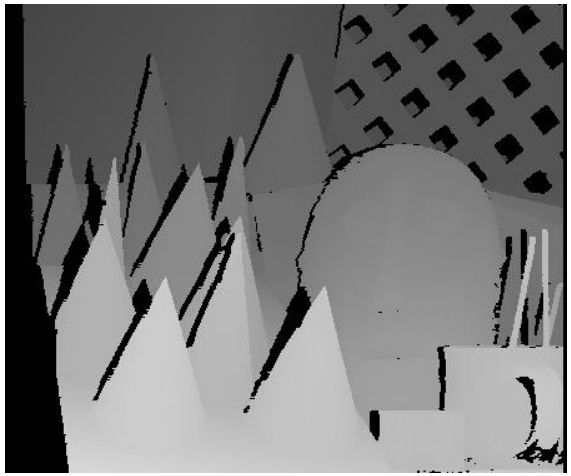
http://vision.middlebury.edu/stereo/



Stereo pair

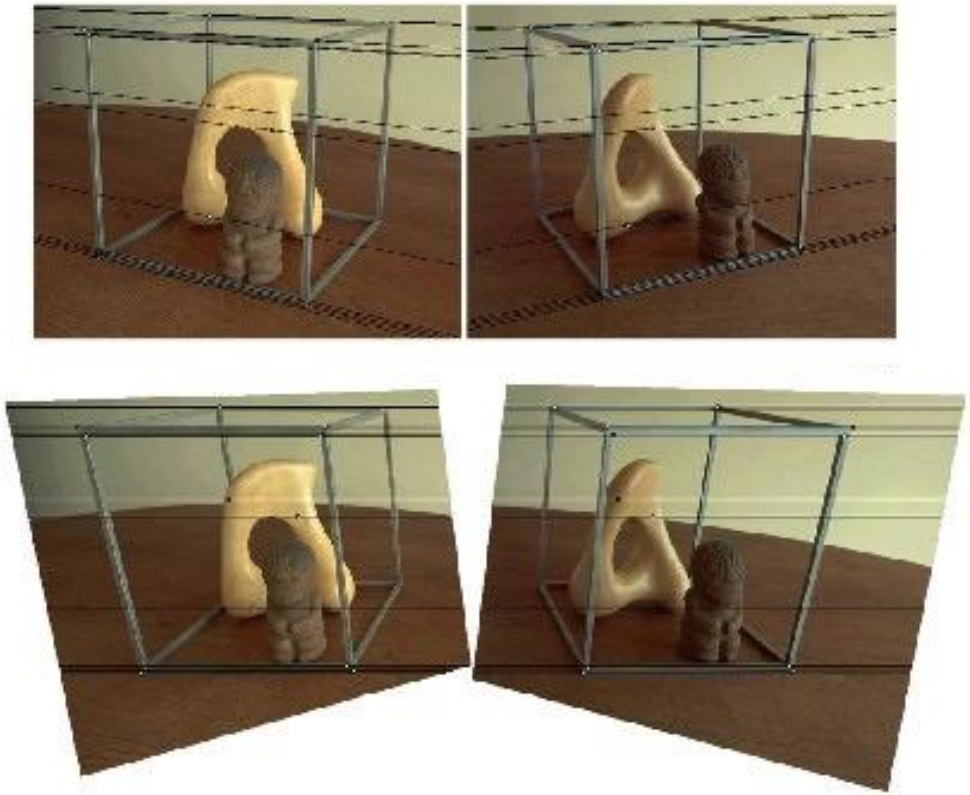
$$p_u - p'_u \propto \frac{B \cdot f}{z}$$

[Eq. 1]



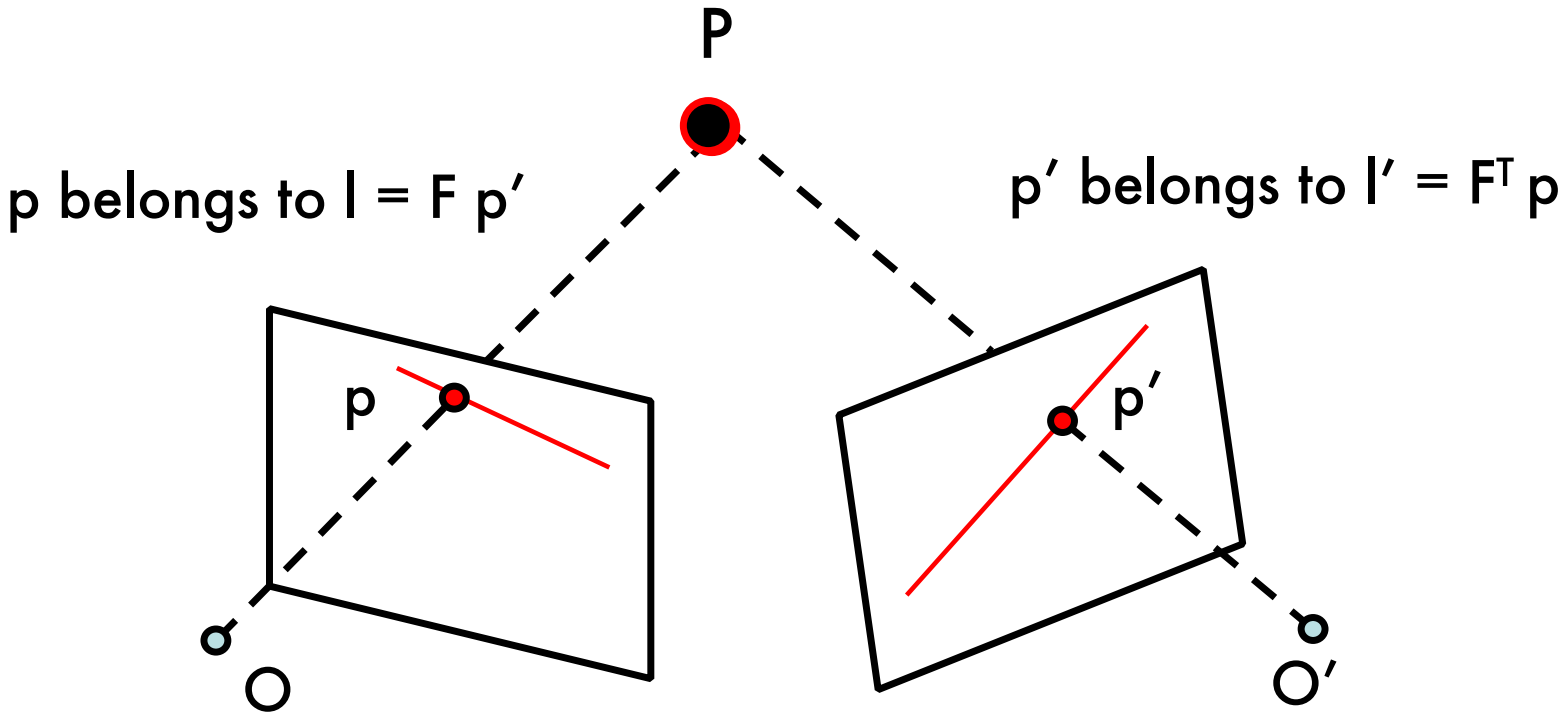
Disparity map / depth map

Why are parallel images useful?



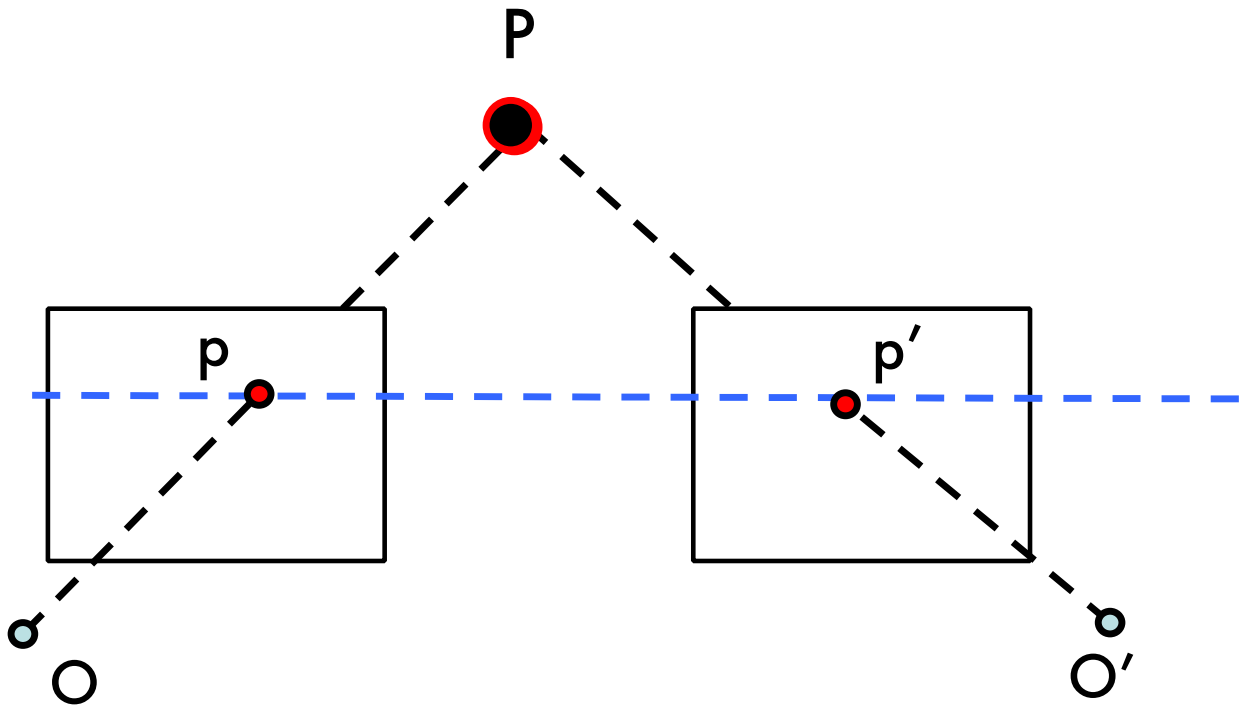
- Makes triangulation easy
- Makes the correspondence problem easier

Correspondence problem



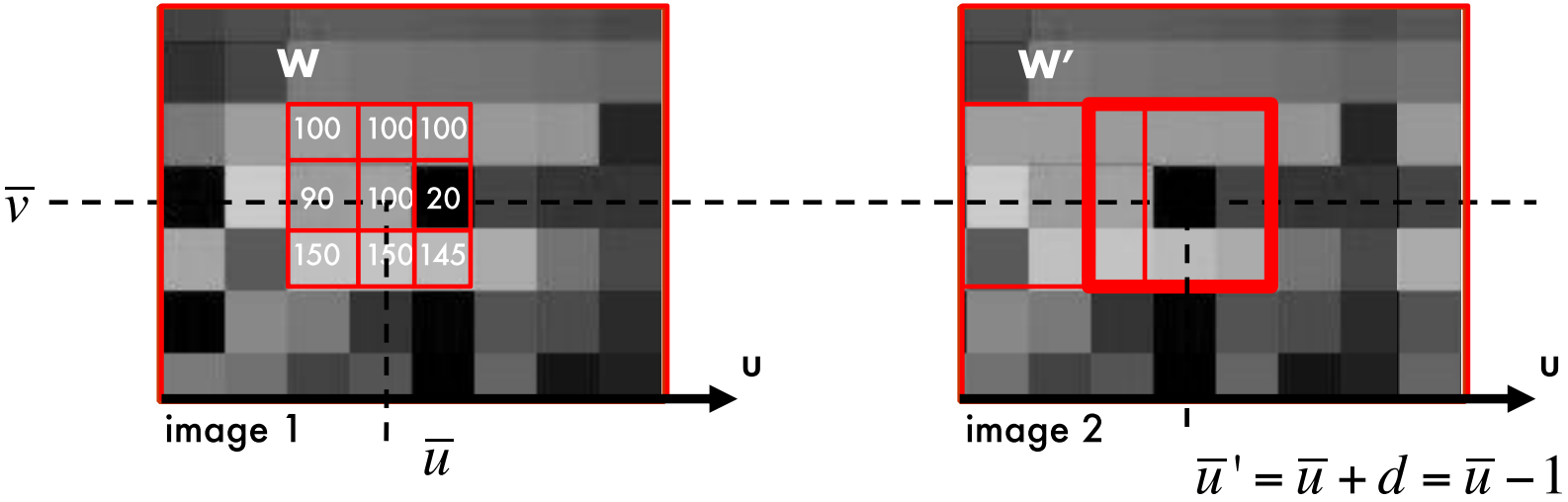
Given a point in 3D, discover corresponding observations in left and right images [also called binocular fusion problem]

Correspondence problem



When images are rectified, this problem is much easier!

Window-based correlation



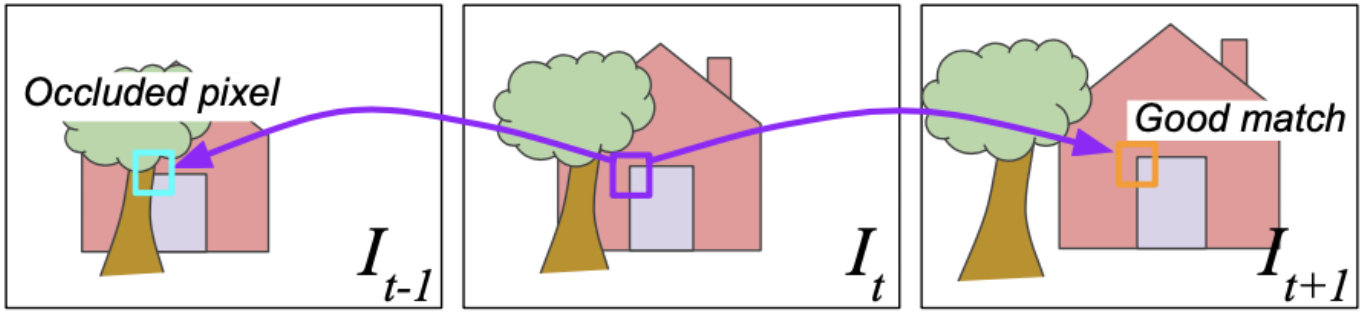
Example: **W** is a 3x3 window in red

w is a 9x1 vector

w = [100, 100, 100, 90, 100, 20, 150, 150, 145]^T

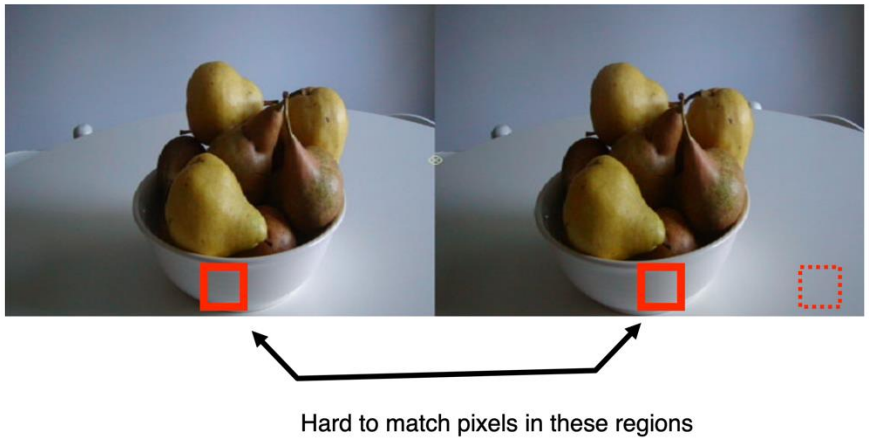
- Pick up a window **W** around $\overline{p} = (\overline{u}, \overline{v})$
- Build vector **w**
- Slide the window **W** along v = $\overline{v}$ in image 2 and compute **w'** (u) for each u
- Compute the dot product $\mathbf{w}^T\,\mathbf{w'}$(u) for each u and retain the max value

The Correspondence Problem



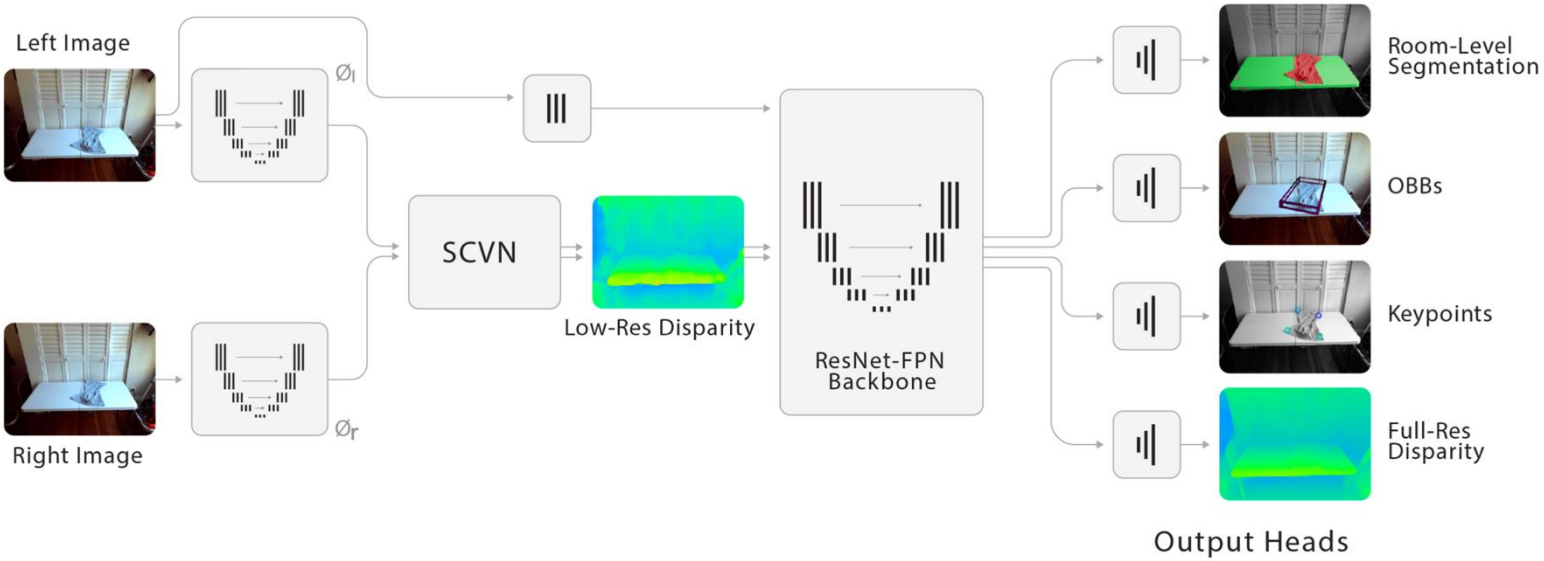
Occlusions

Repetitive Patterns



Homogenous regions

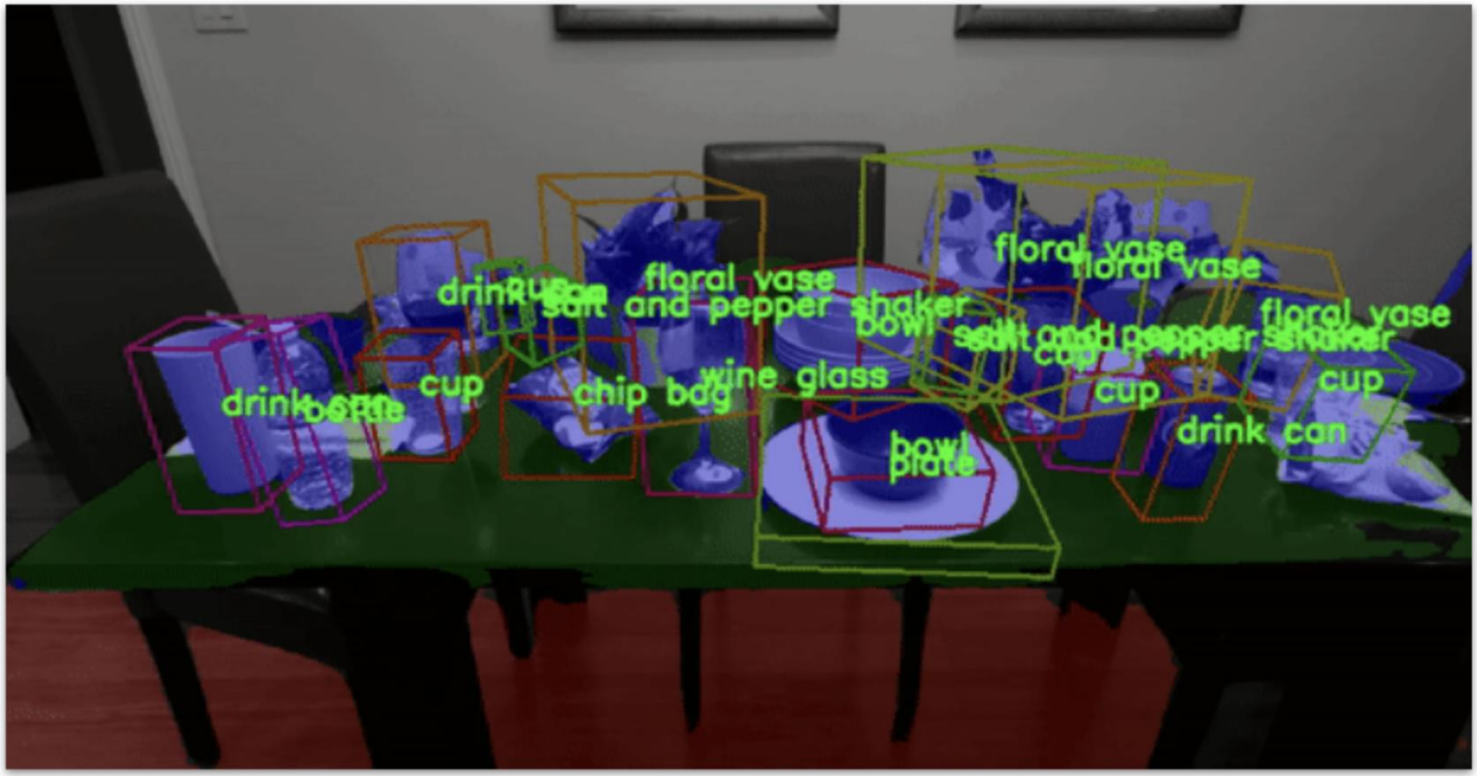
Can we learn a similarity function to find corresponding points?



Supervised Learning of Disparity map.

SimNet: Enabling Robust Unknown Object Manipulation from Pure Synthetic Data via Stereo. CoRL '21. Kollar et al.

3D Scene Understanding



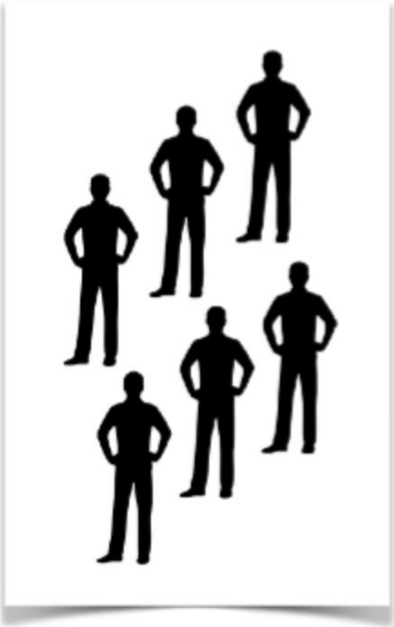
The robot needs to have at least:
-3D detections
-Room Level Segmentation
-Semantic Class Categories

Scaling Up Data Annotation

Buy Labeled Real Data

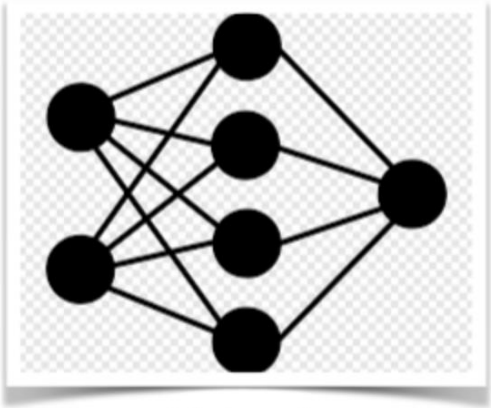
Buy Photorealistic Synthetic Scenes

Challenges with Data Annotation



External Data Contractors

ML Engineer



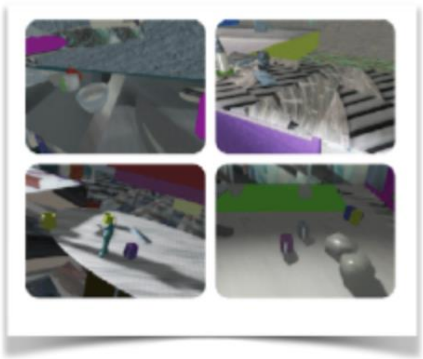
Network

Iteration between contractors/engineers is quite painful

- >100K (USD) per cycle,
- >1 month lead time

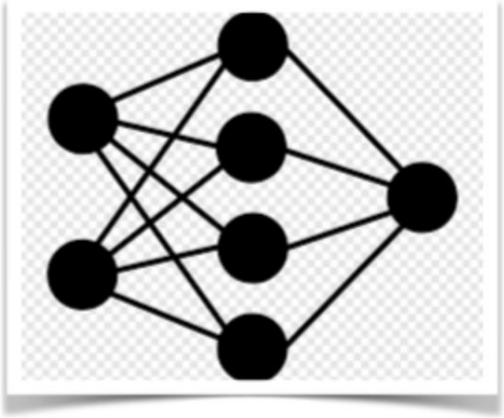
This hinders model prototyping and prevent progress.

What if data was programmable?



Procedural Simulator

ML Engineer



Network

Image = Geometry + Appearance

An image is primarily composed of:

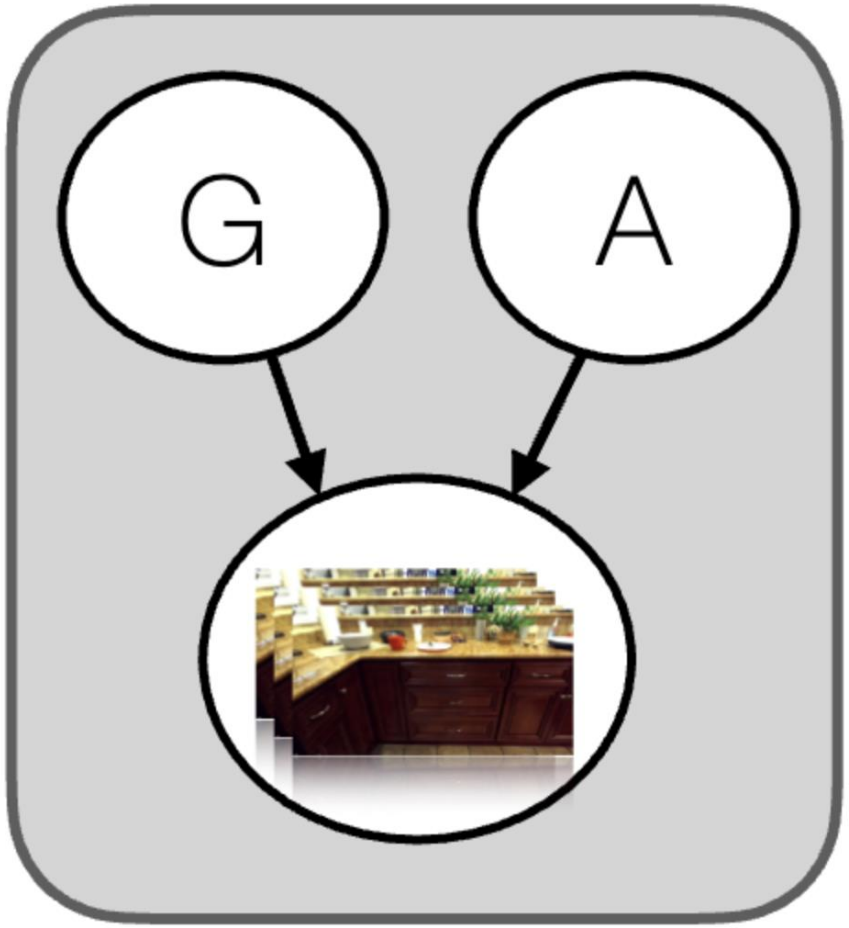
1. **G**eometry
2. **A**ppearance - materials, texture, lighting

Hypothesis:

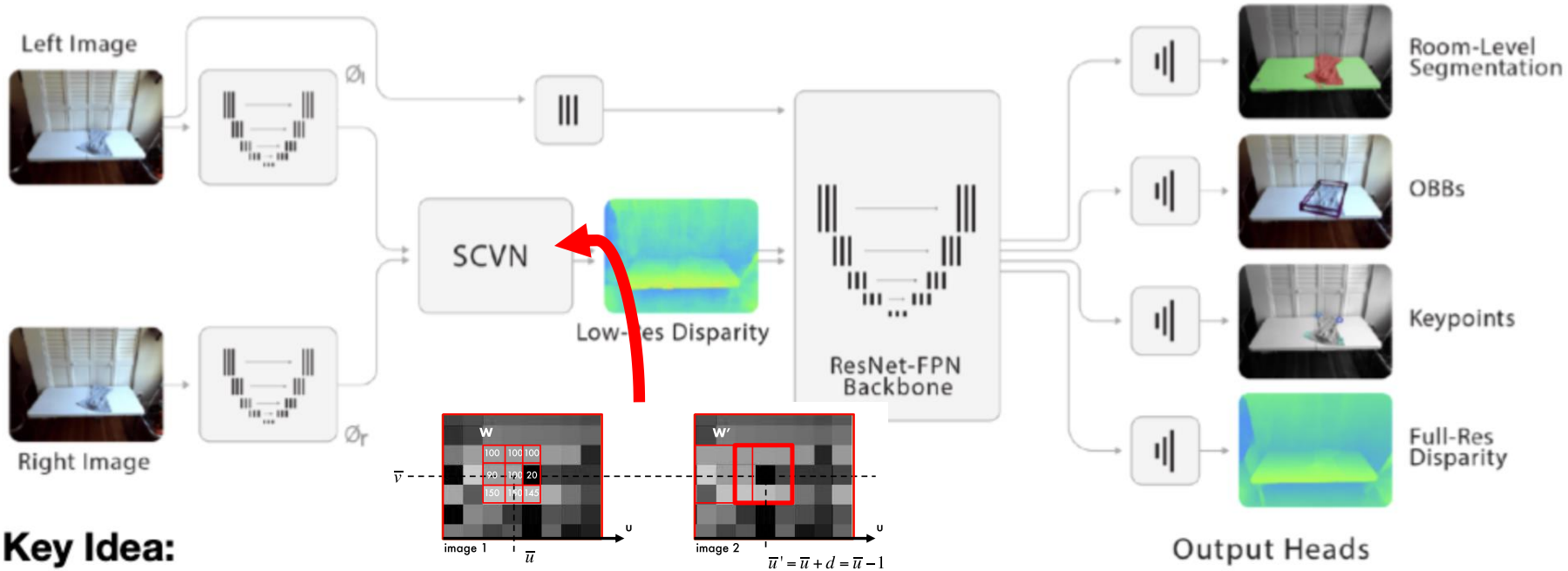
- Geometry **easy** to render/model
- Appearance **hard** to render/model

To achieve sim-to-real transfer, we need to:

1. Minimize the use of appearance features
2. Randomize over scene geometry



Minimize
Use of Appearance Features



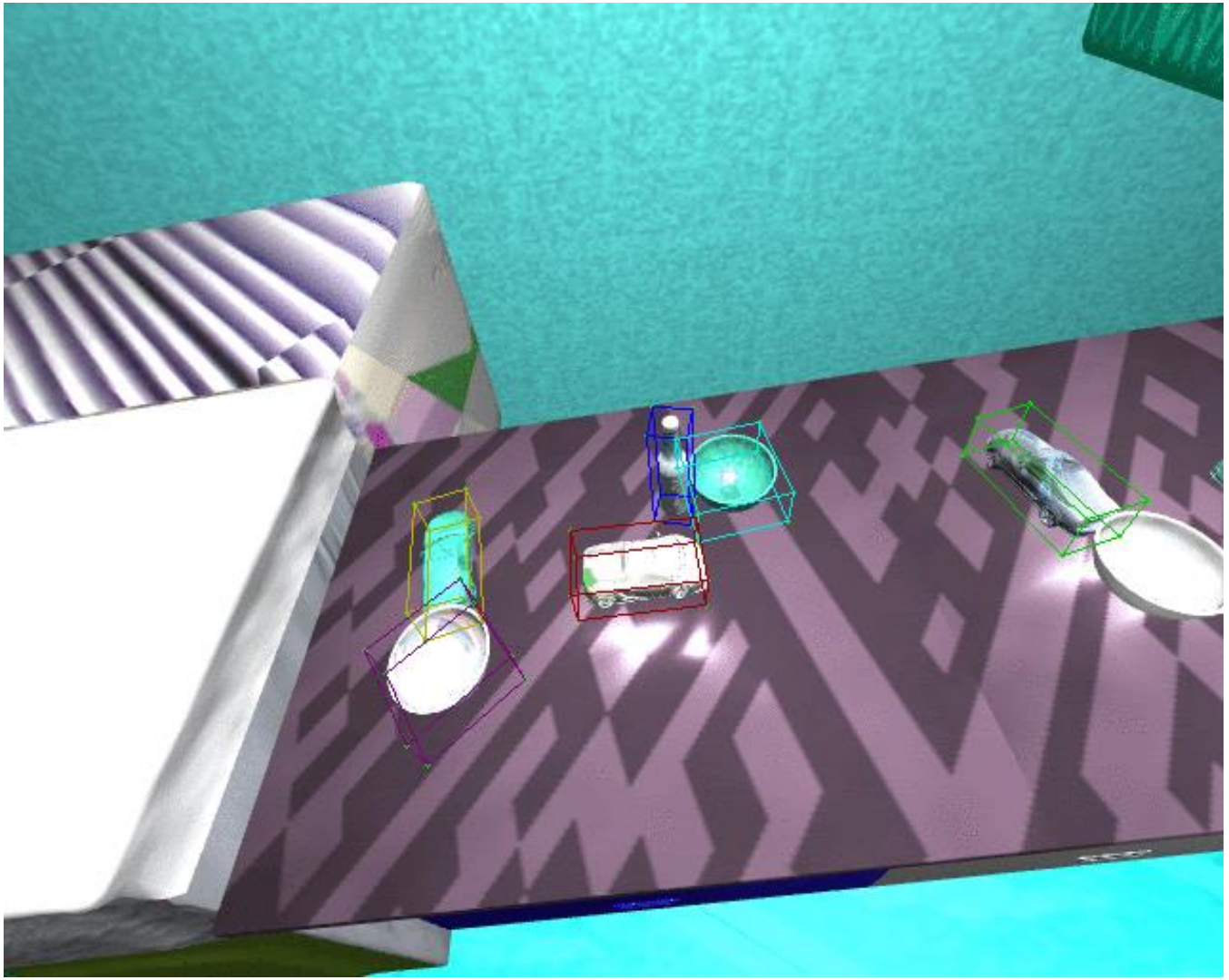
Key Idea:
Add *explicit stereo reasoning* in the network to extract geometric features during inference.

$$c(\mathbf{x}_1, \mathbf{x}_2) = \sum_{\mathbf{o} \in [-k,k] \times [-k,k]} \langle \mathbf{f}_1(\mathbf{x}_1 + \mathbf{o}), \mathbf{f}_2(\mathbf{x}_2 + \mathbf{o}) \rangle$$

$\boldsymbol{f_1}$ = Feature
x_1 = Image Patch location
k = ½ Image Patch dimension

Randomize over Geometry

Densely Annotated Scenes



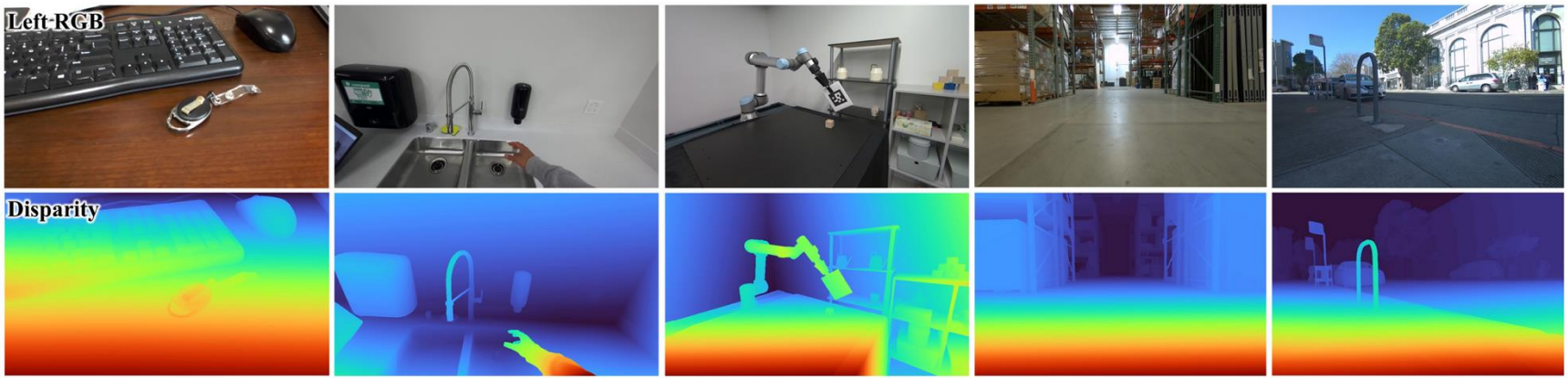
3D Scene Understanding

TODO: Foundation Stereo

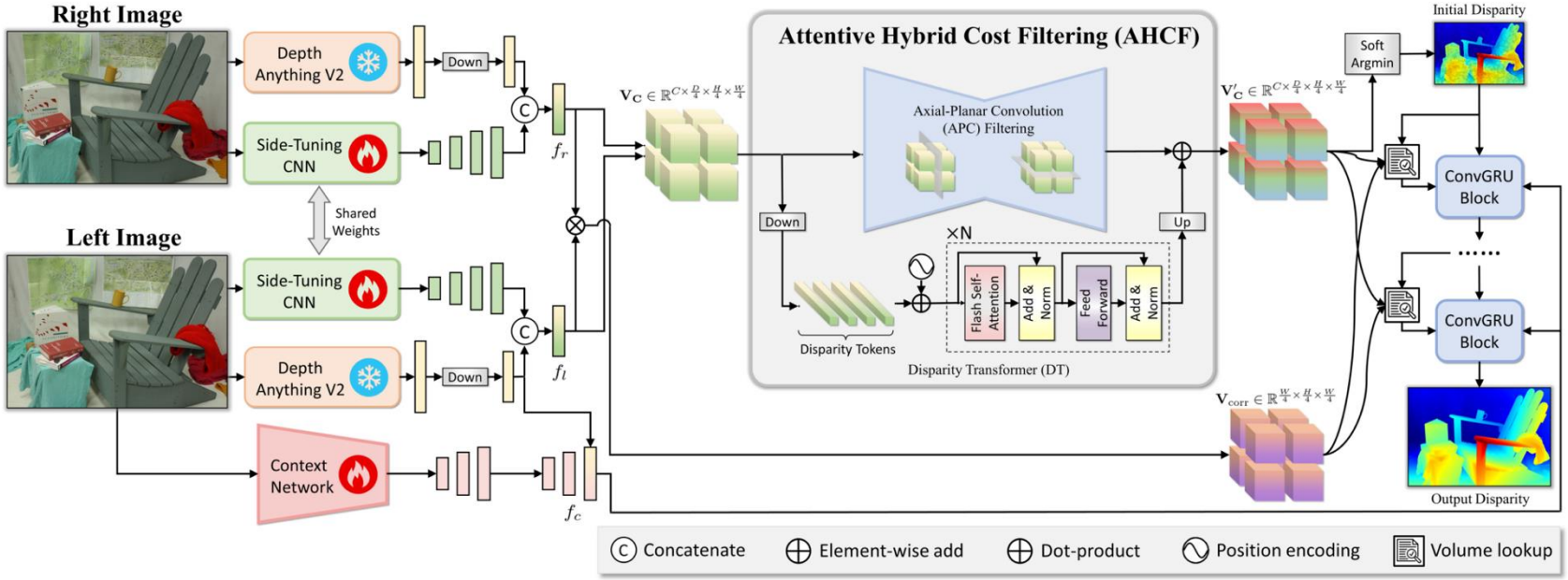
- Focus on correlation portion within Foundation Stereo
- Synthetic images for training
- What is different from earlier work?
  - Much more complex architecture and use of DepthAnything priors from monucular stereo

FoundationStereo: Zero Shot Stereo Matching. Wen et al. CVPR 2025.

- Synthetic Training Images
- Much more complex architecture
- At the core: Feature Correlation
- Here: Group-wise Correlation



Foundation Stereo

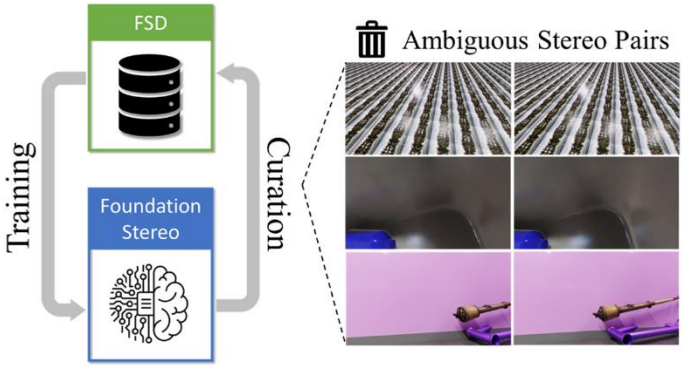


$$\mathbf{V}_{\text{gwc}}(g,d,h,w) = \left\langle \widehat{f}^{(4)}_{l,g}(h,w), \widehat{f}^{(4)}_{r,g}(h,w-d) \right\rangle,$$

$$\mathbf{V}_{\text{cat}}(d,h,w) = \left[\text{Conv}(f^{(4)}_{l})(h,w), \text{Conv}(f^{(4)}_{r})(h,w-d) \right]$$

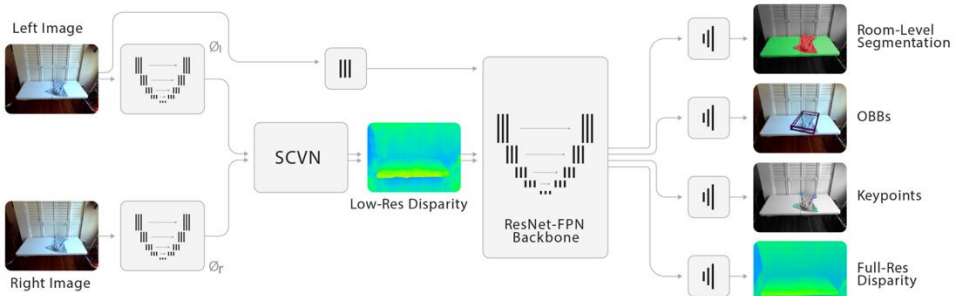
$$\mathbf{V}_{\mathbf{C}}(d,h,w) = [\mathbf{V}_{\text{gwc}}(d,h,w), \mathbf{V}_{\text{cat}}(d,h,w)] \qquad (1)$$

Foundation Stereo

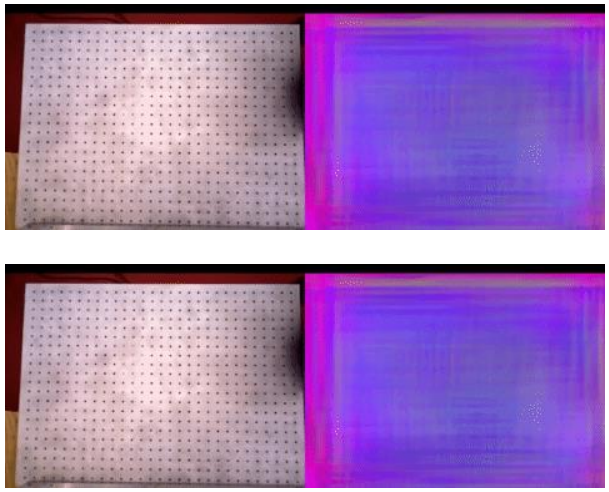


- Synthetic Training Images
- With many textures, geometries and poses
- Self-curation by discarding image pairs that are hard to predict

Let's use representation learning!



Depth Estimation from Stereo
Supervised Learning



Monocular Depth Estimation
Unsupervised Learning

Image by Yunuk Cha.

Finding Correspondences across Frames
Self-Supervised Learning

What if we don’t need to form correspondences between images?

- Can we estimate depth from a single image?

Can you estimate depth from a single RGB image?

Yes

No

Can you estimate depth from a single RGB image?

Yes

0%

No

0%

Can you estimate depth from a single RGB image?

Yes

0%

No

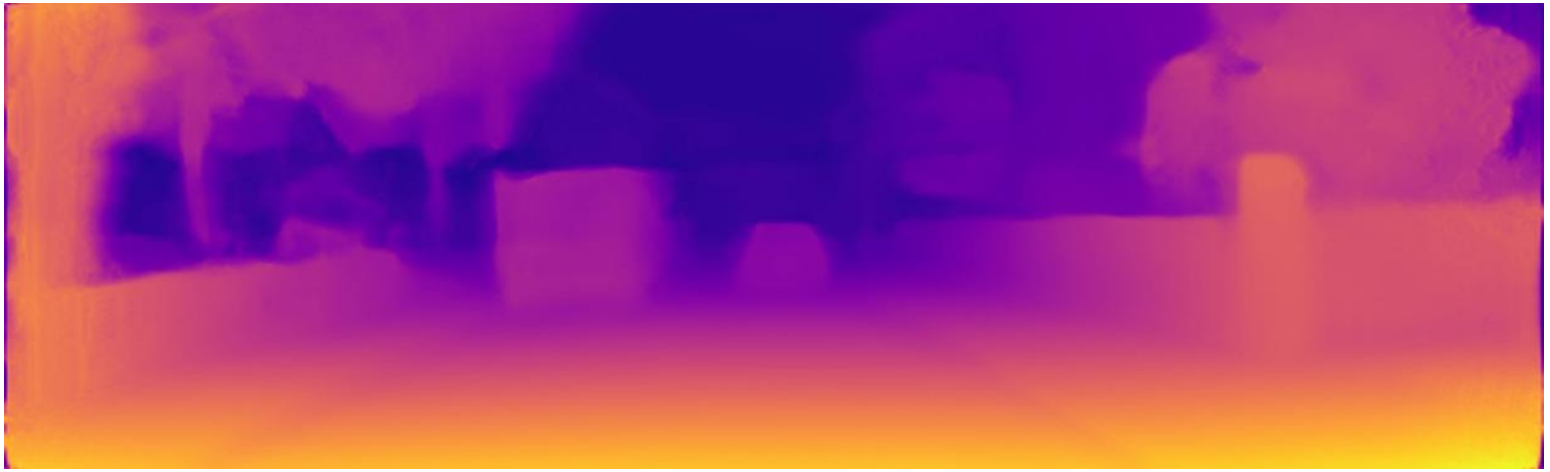
0%

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

Input image

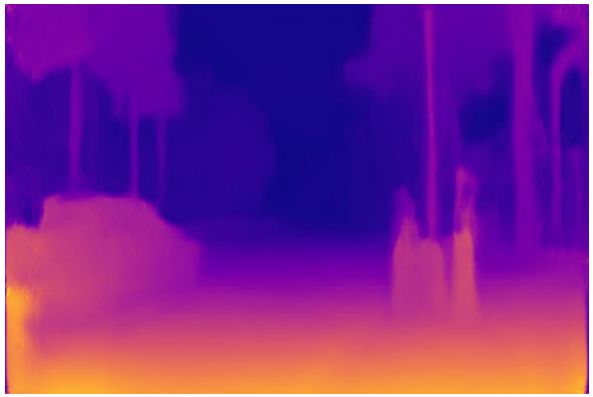


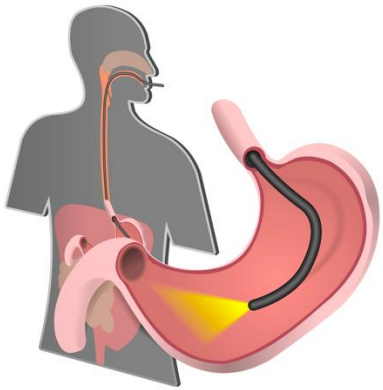
Result

Why depth?

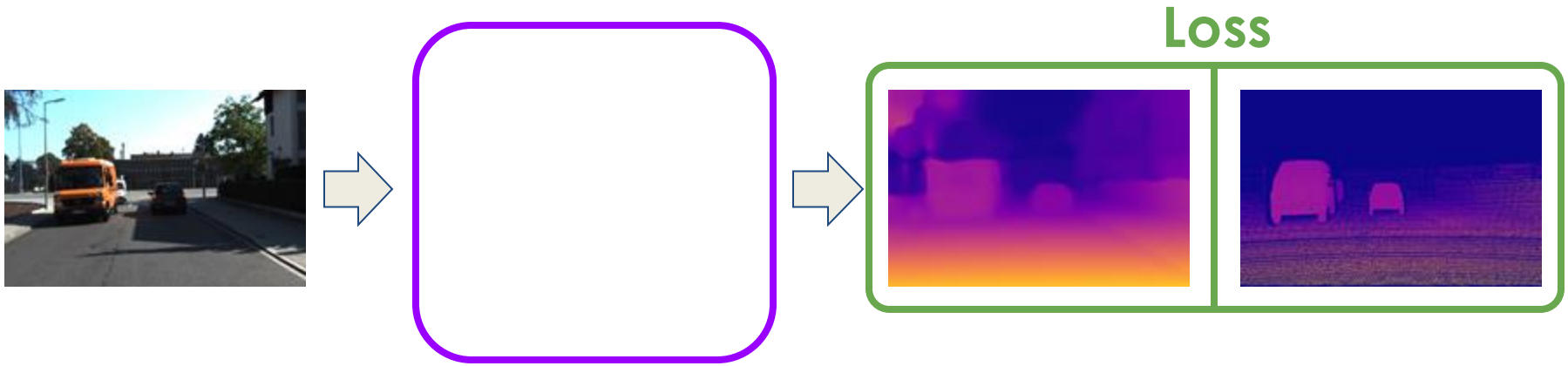
How do we usually get depth?

Why monocular?



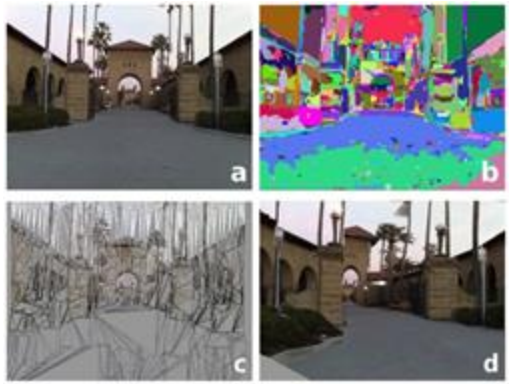


Previous approaches - Supervised



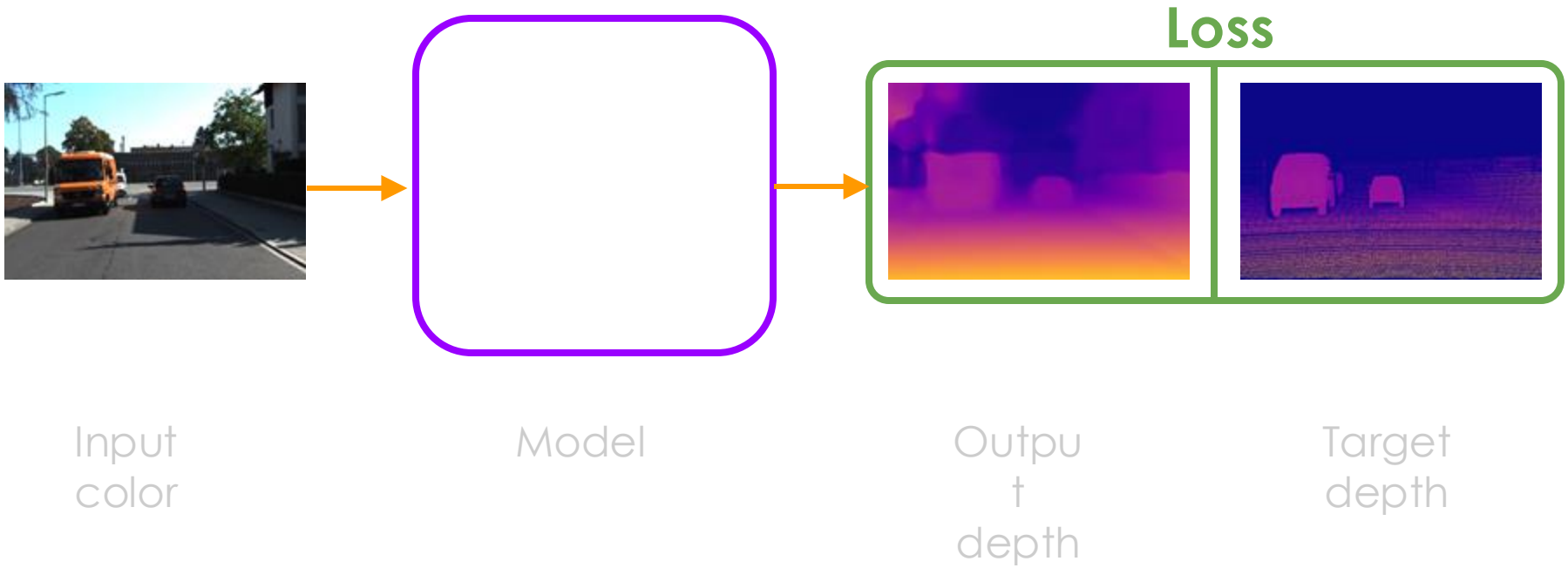
Previous approaches - Supervised

Automatic Photo Pop-up [SIGGRAPH 05]

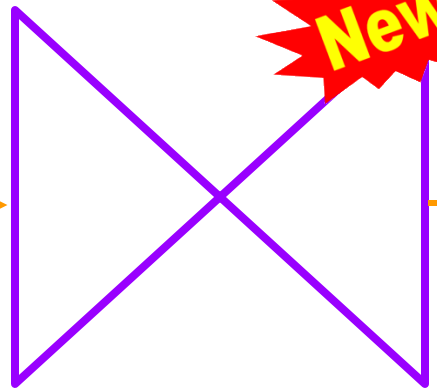


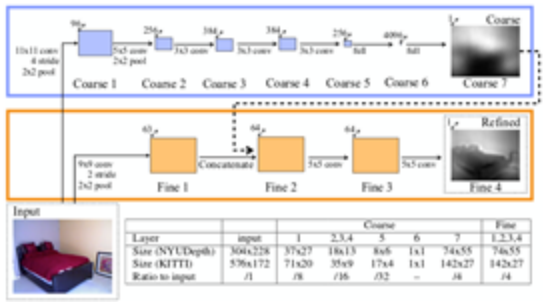
Make3D [PAMI 08]

Previous approaches - Supervised



Previous approaches - Supervised





Input color

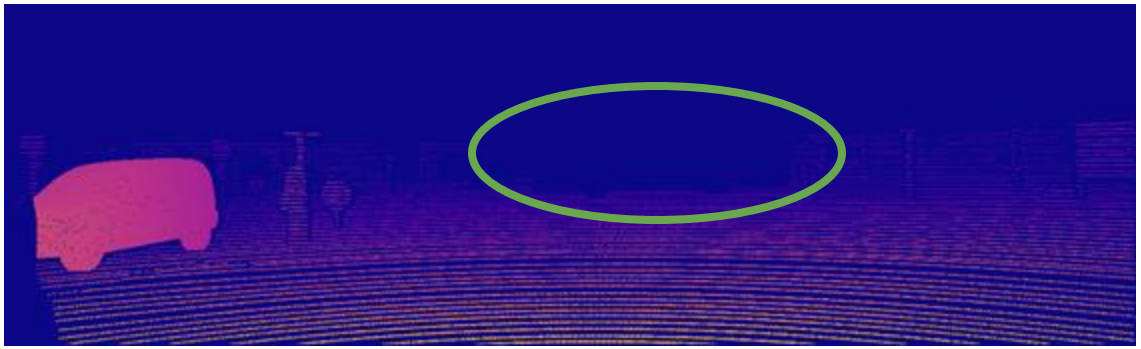
Output depth

Target depth

Eigen et al. [NIPS 14]
Li et al., Laina et al., Cao et al., ...

KITTI 2015

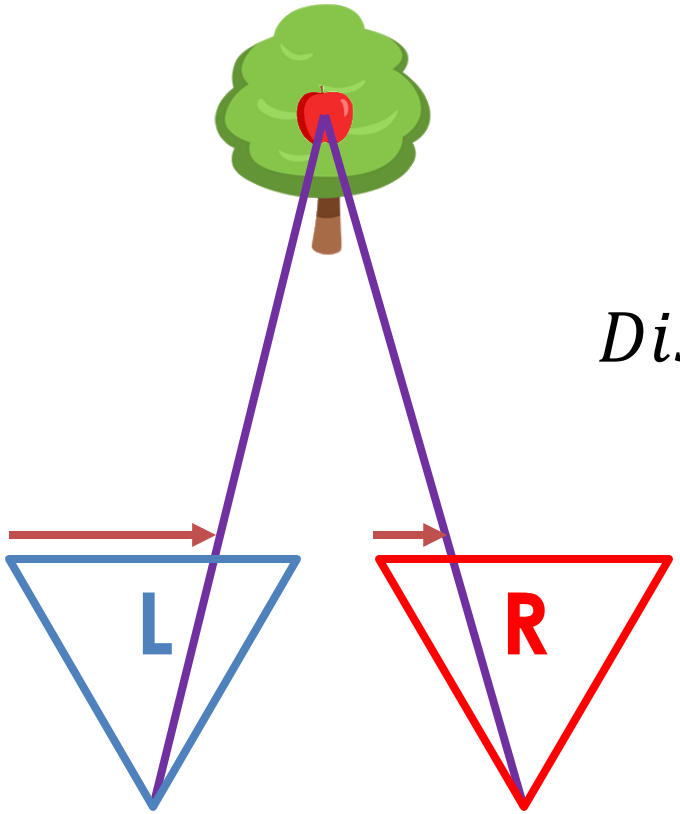

Velodyne HDL-64E Laserscanner
Point Gray Flea 2
Video Cameras
AnnieWAY
KIT
KA EV 842



IR Structured Light

- Does not work well outside

Depth from stereo



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

Let's train with stereo data!

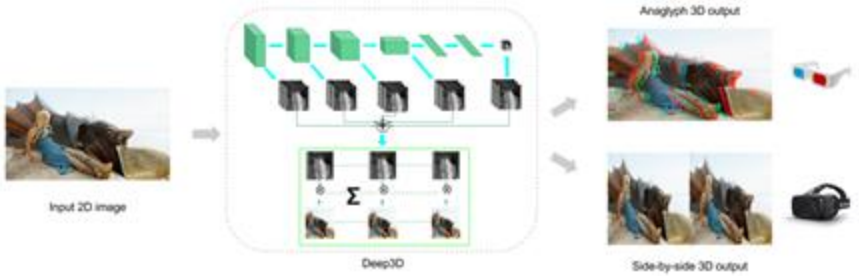
Point Grey

Apple

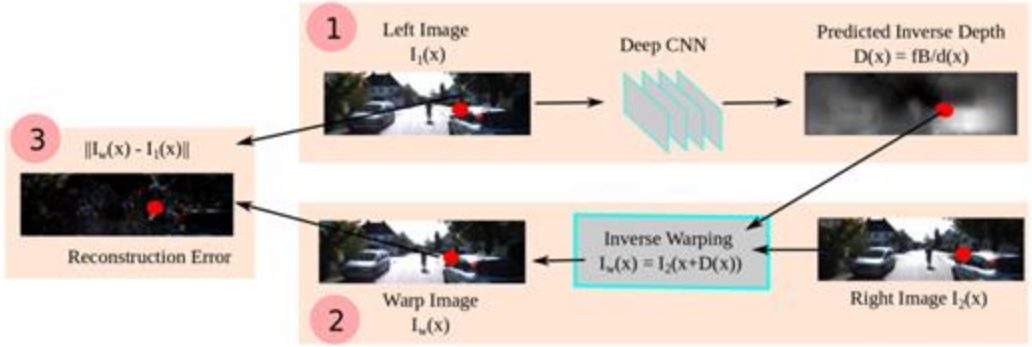
Stereolabs

Previous approaches - Unsupervised

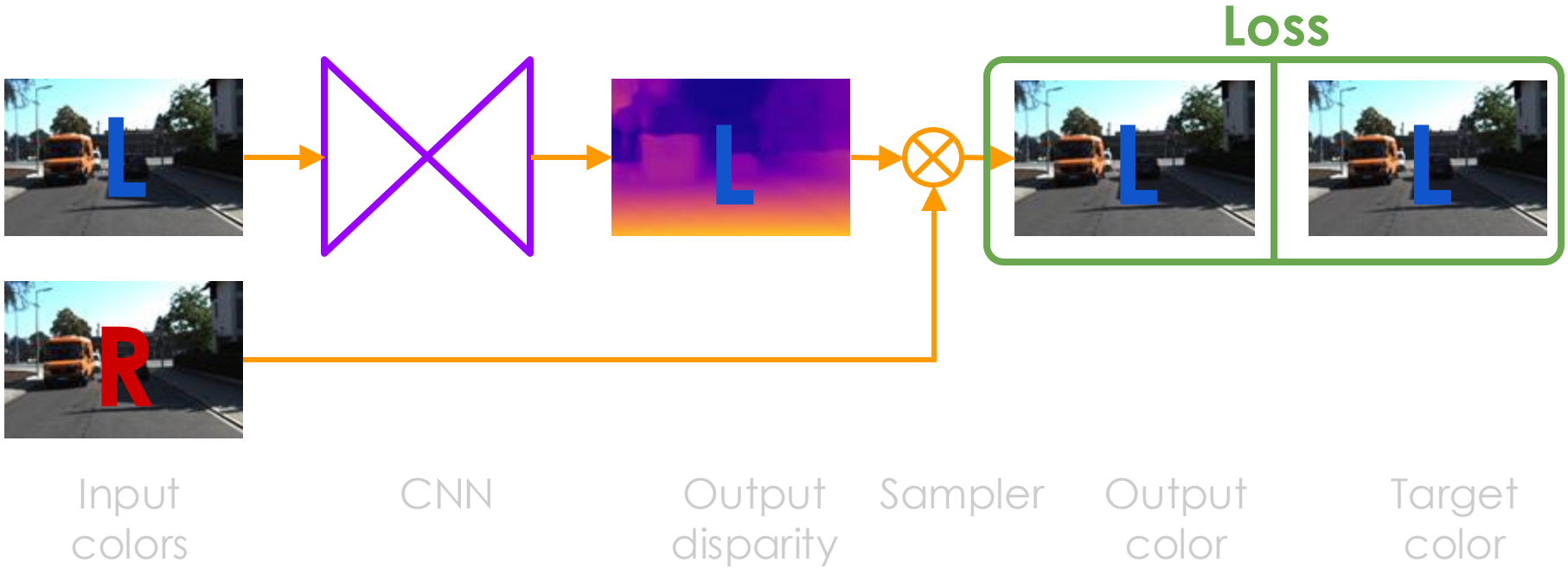
Deep3D
Xie et al. [ECCV 16]



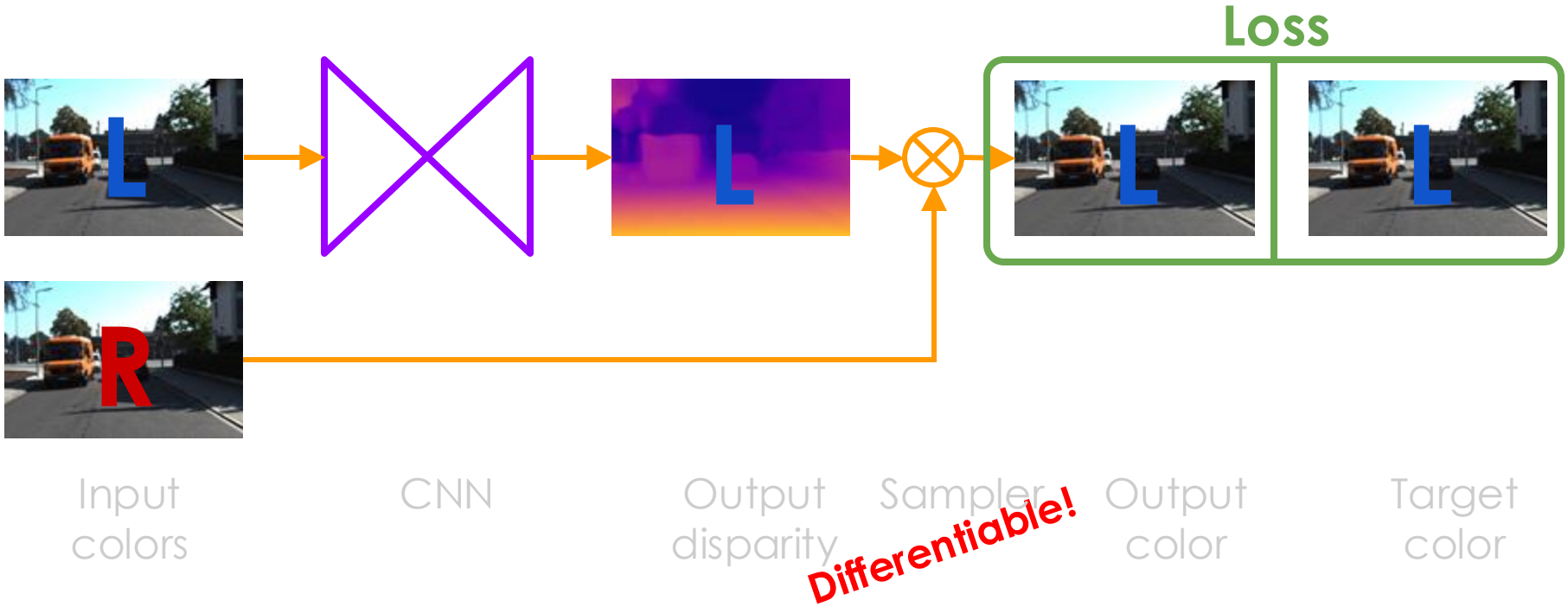
Garg et al. [ECCV 16]



Unsupervised depth estimation - Concept

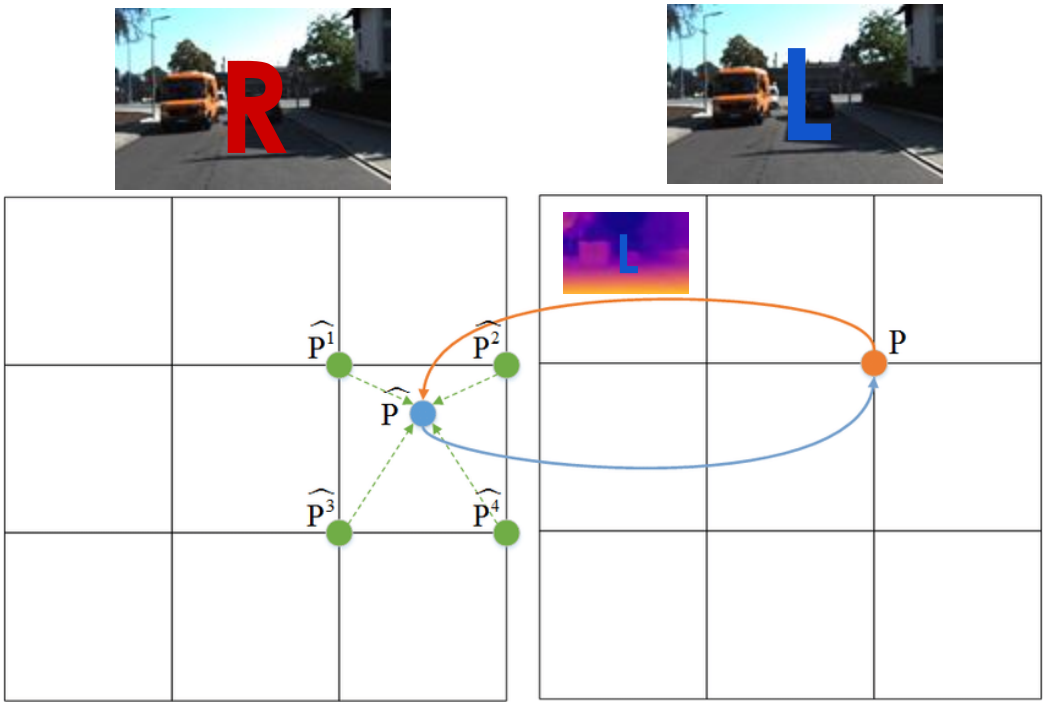


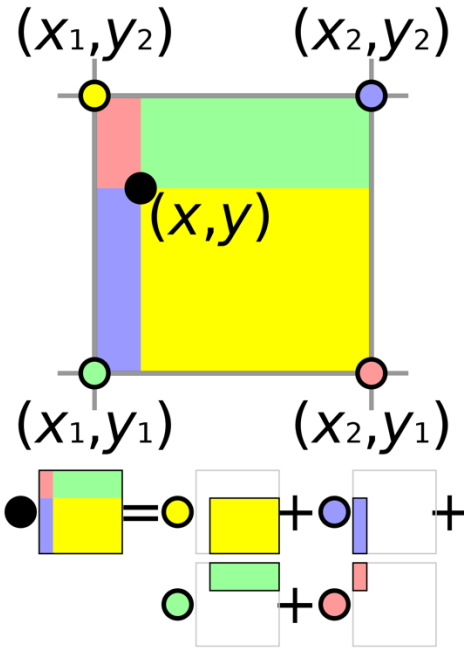
Unsupervised depth estimation - Baseline



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

Bilinear Sampling



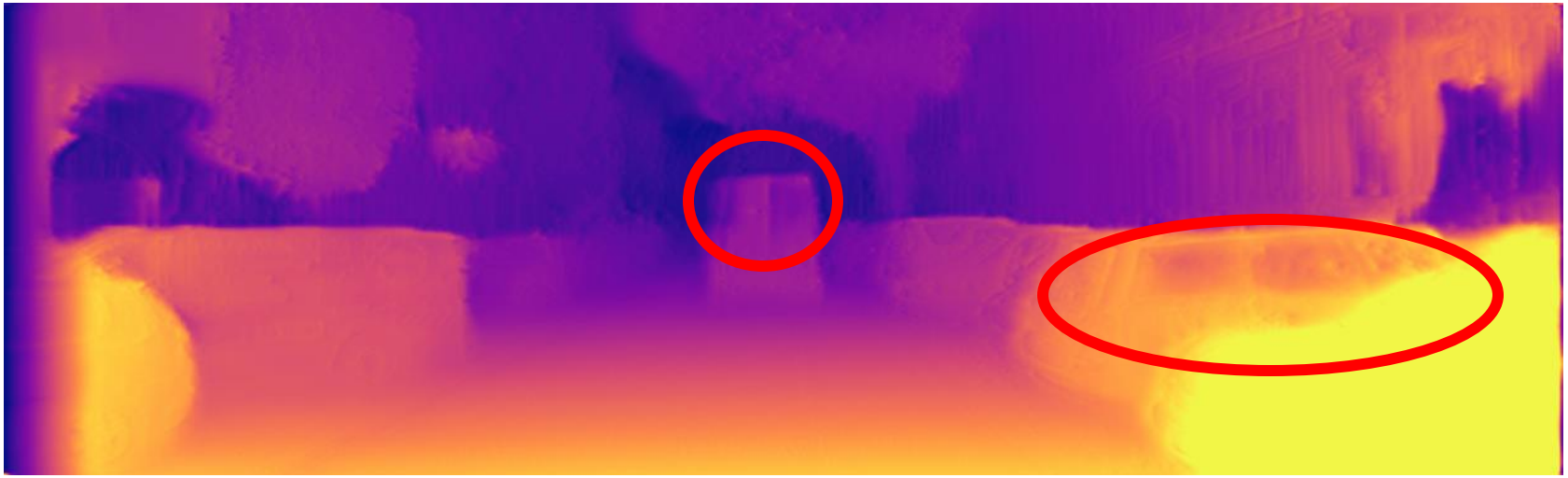


Spatial transformer networks, Jaderberg et al. [NIPS 15]

Input

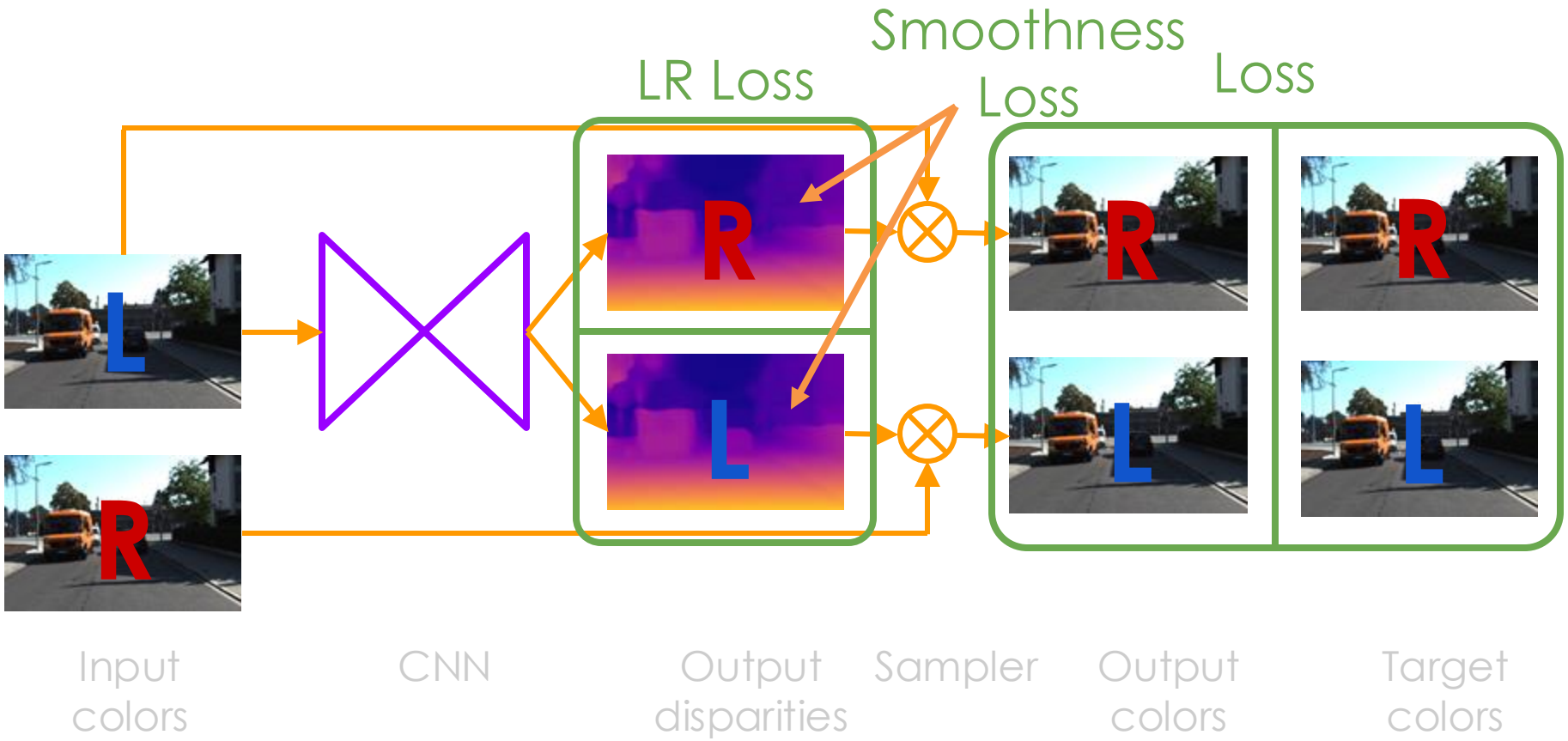
Baseline

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

N = # of pixels
i, j = index of the pixels
α = weight

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

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

N = # of pixels
i, j = index of the pixels
d = disparity in left/right image

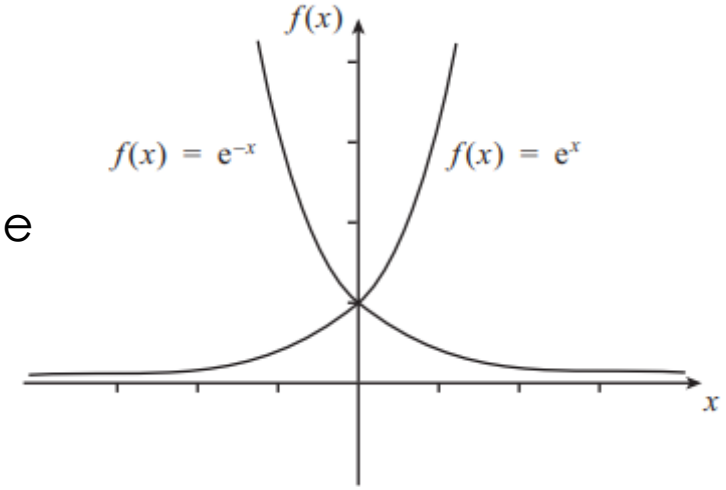
$$C_{lr}^l = \frac{1}{N} \sum_{i,j} \left| d_{ij}^l - d_{ij+d_{ij}^l}^r \right|$$

Smoothness Loss

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

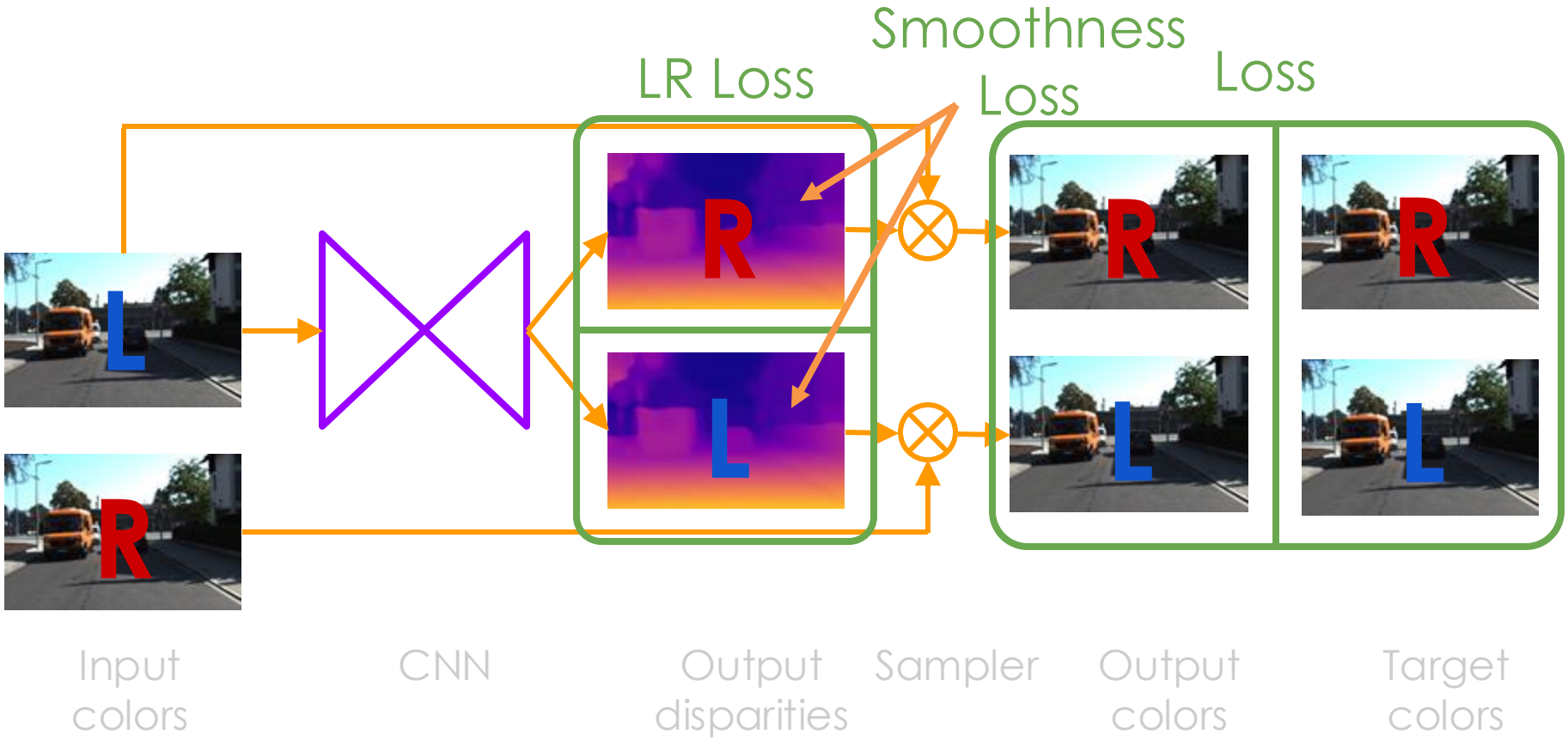
$$C_{ds}^l = \frac{1}{N}\sum_{i,j} |\partial_x d_{ij}^l| e^{-\|\partial_x I_{ij}^l\|} + |\partial_y d_{ij}^l| e^{-\|\partial_y I_{ij}^l\|}$$

N = # of pixels
i, j = index of the pixels
d = disparity in left/right image
$\partial_x d_{i,j}^l$ = Gradient of disparity
$\partial_x I_{i,j}^l$ = Image Gradient



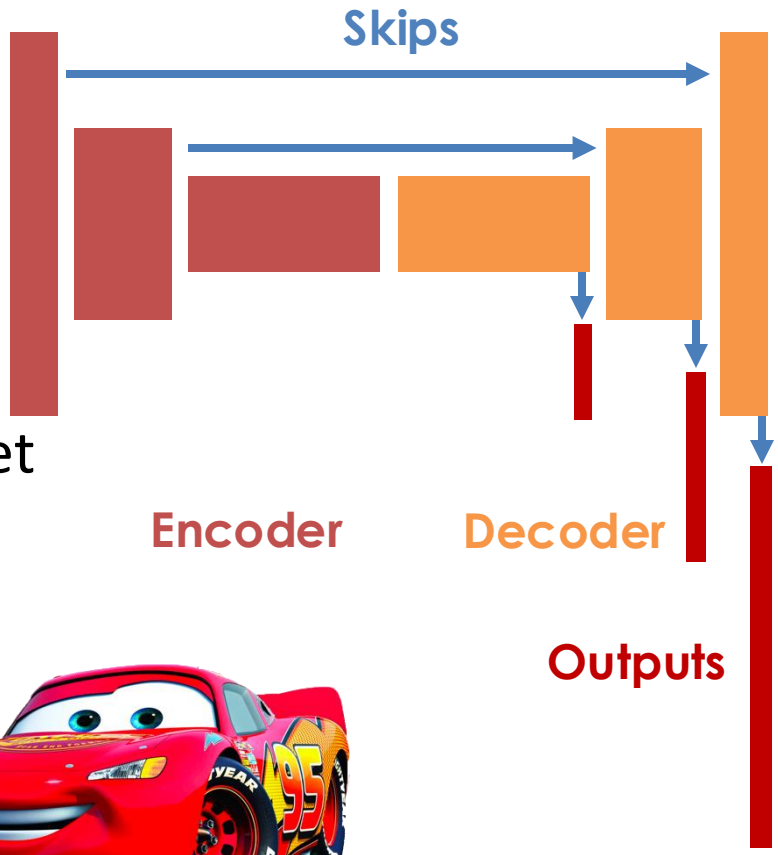
Unsupervised depth estimation - This method

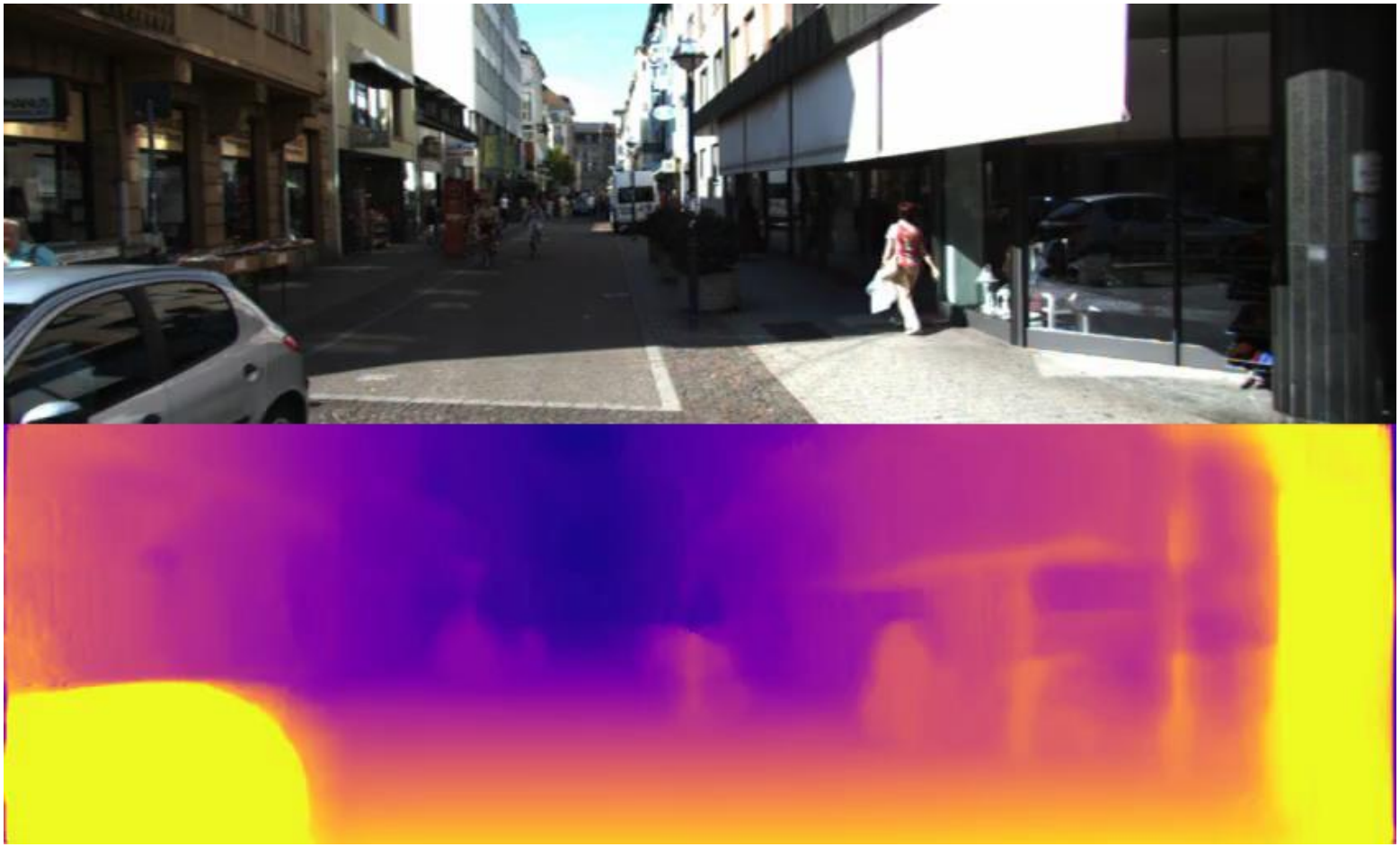
$$C_s = \alpha_{ap}(C_{ap}^l + C_{ap}^r) + \alpha_{ds}(C_{ds}^l + C_{ds}^r) + \alpha_{lr}(C_{lr}^l + C_{lr}^r)$$



Architecture

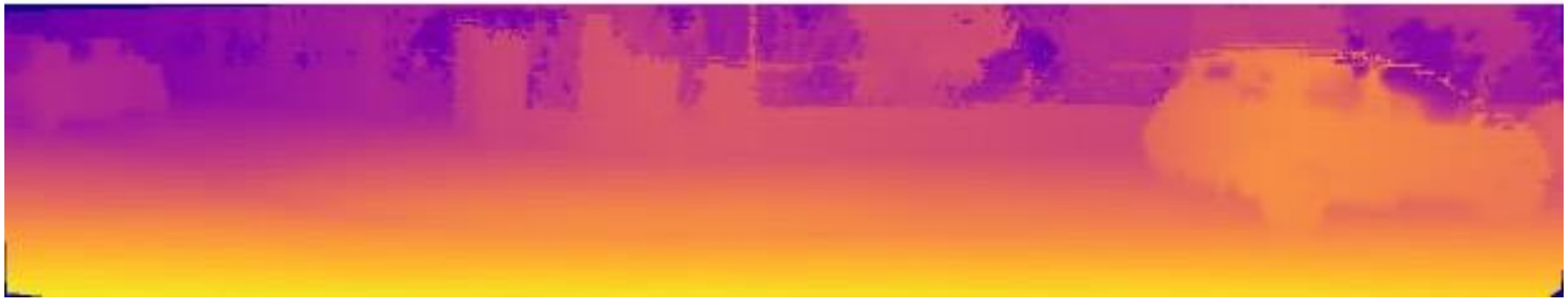
- Fully convolutional
  - Choose your favorite encoder
- Skip connections
  - Similar to DispNet and FlowNet
- Multiscale generation
  - And Loss!
- Fast!
  - ~30fps on a Titan X



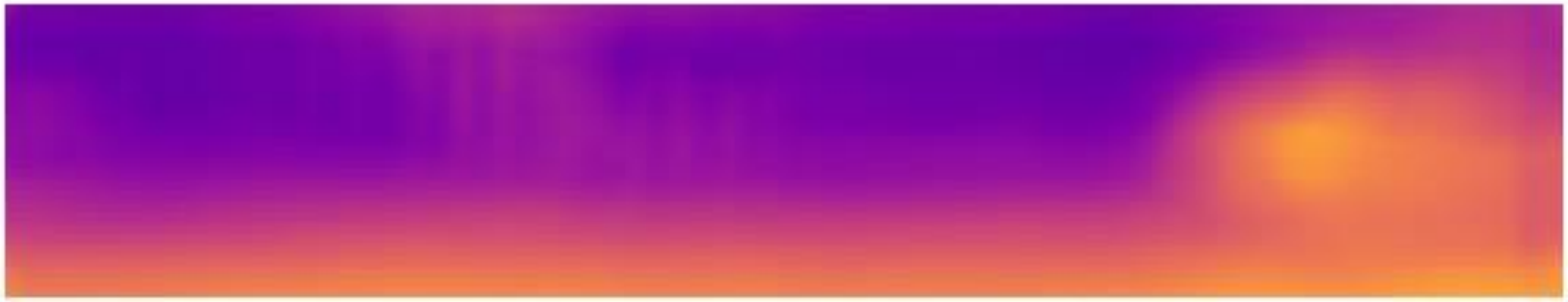


KITTI – Input image

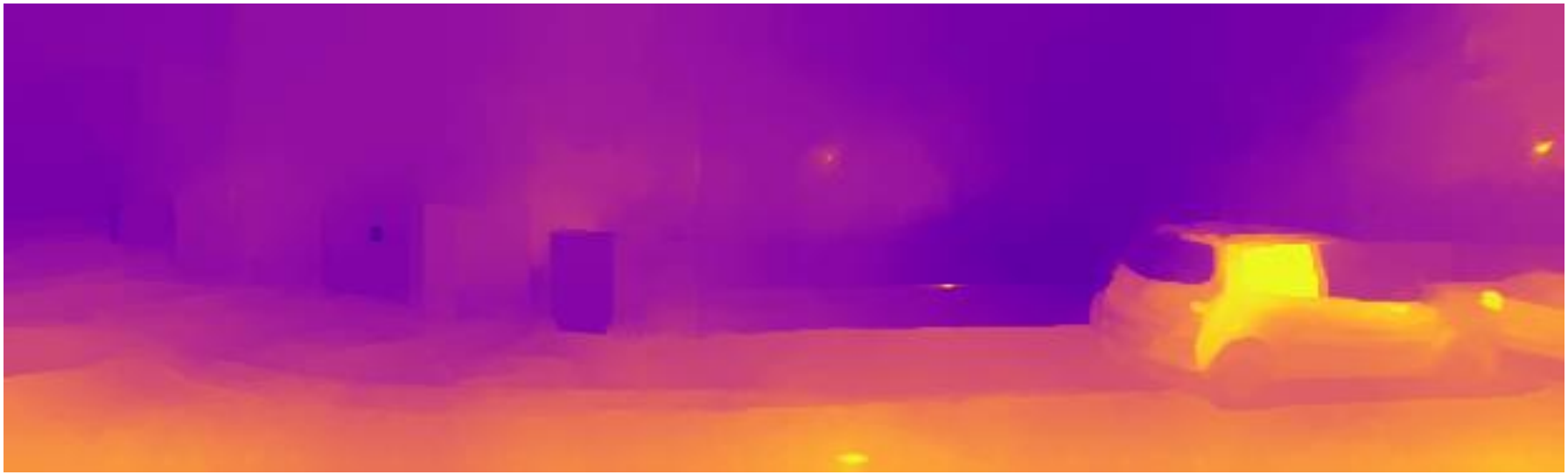
KITTI – Ground truth depth



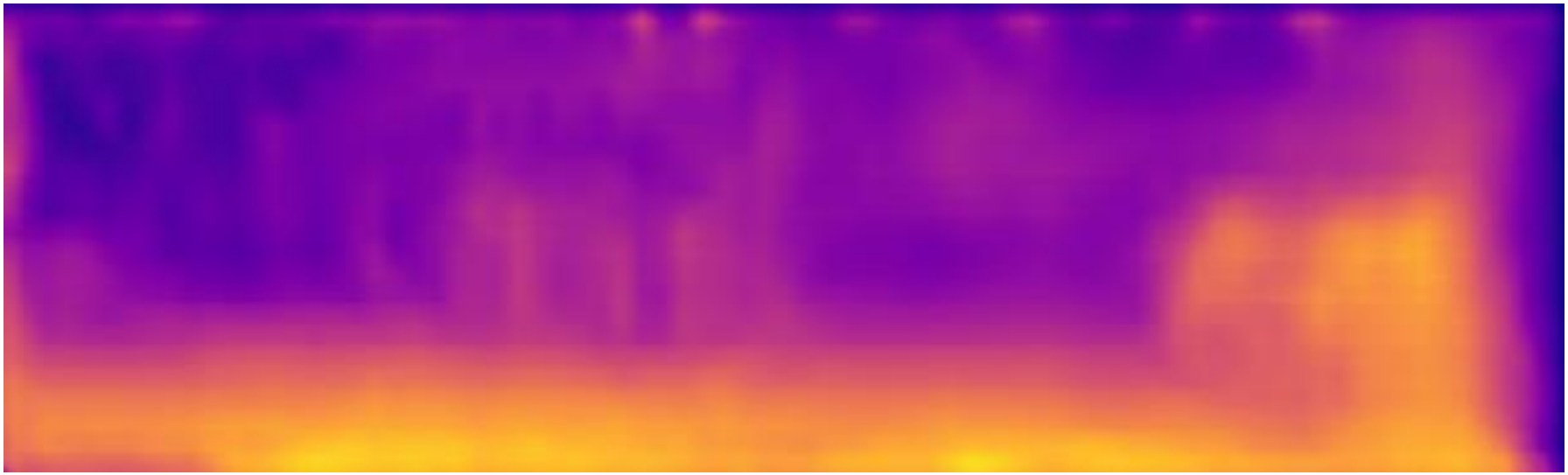
KITTI – Eigen *et al.* [NIPS 14]



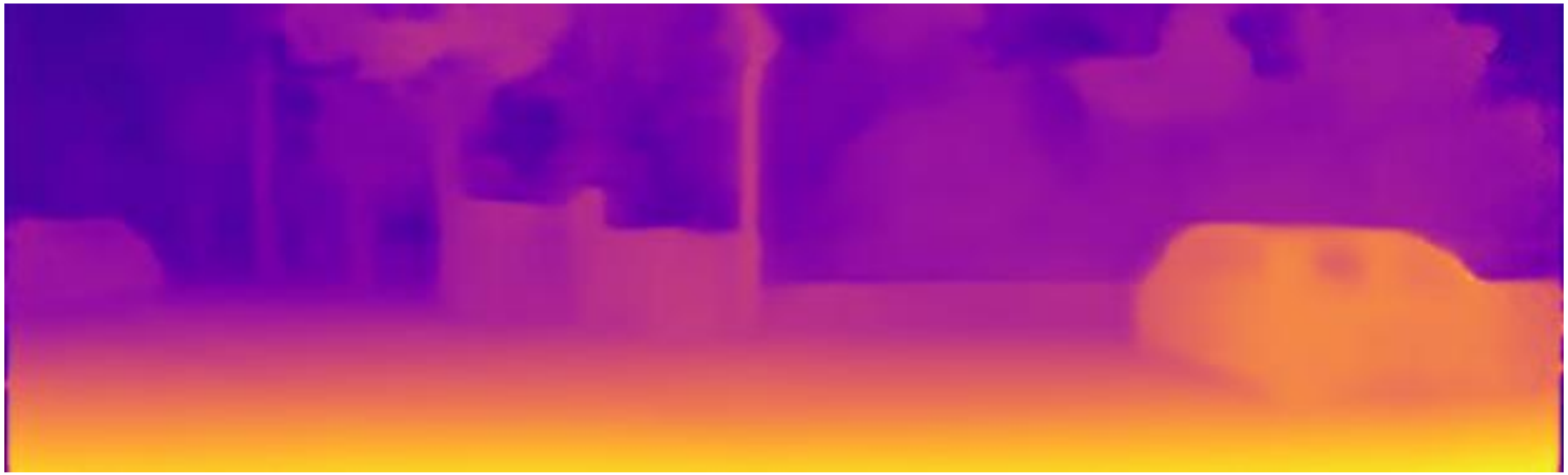
KITTI – Liu *et al.* [CVPR 14]



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

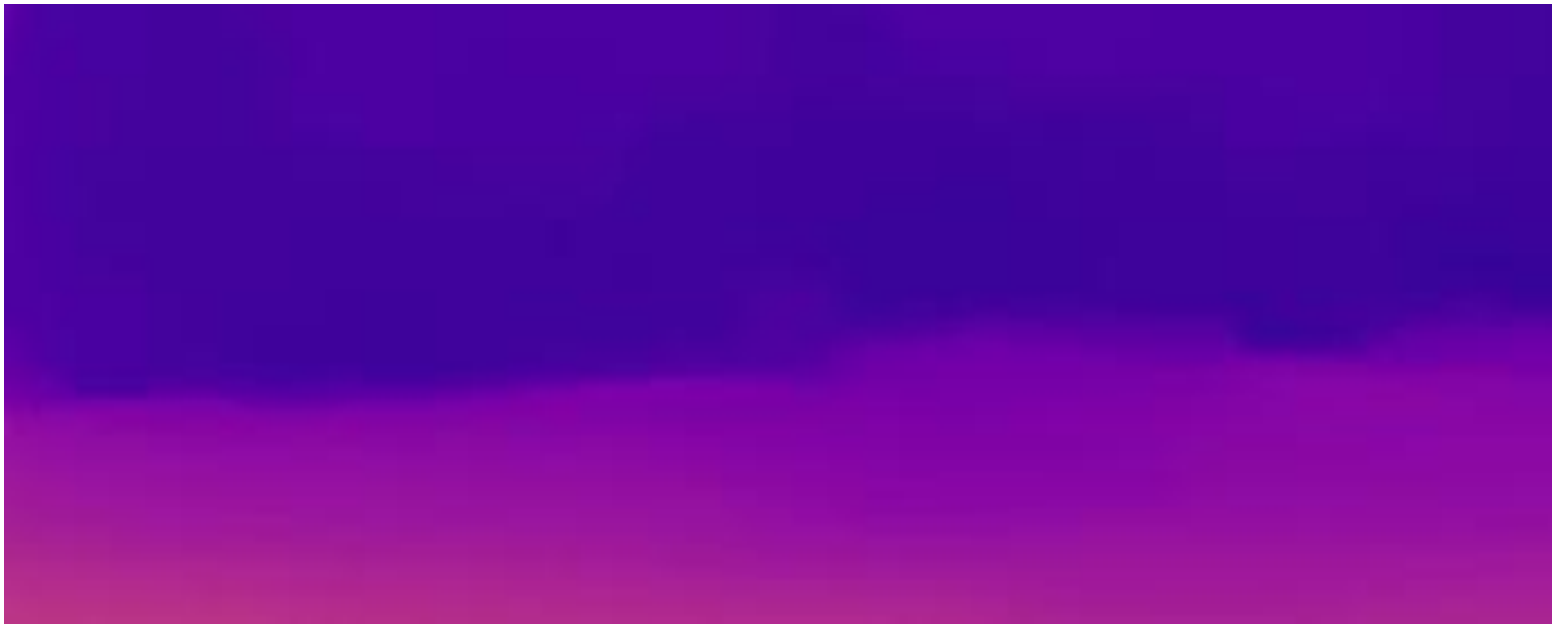
- All variants of our model beat the previous unsupervised method

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

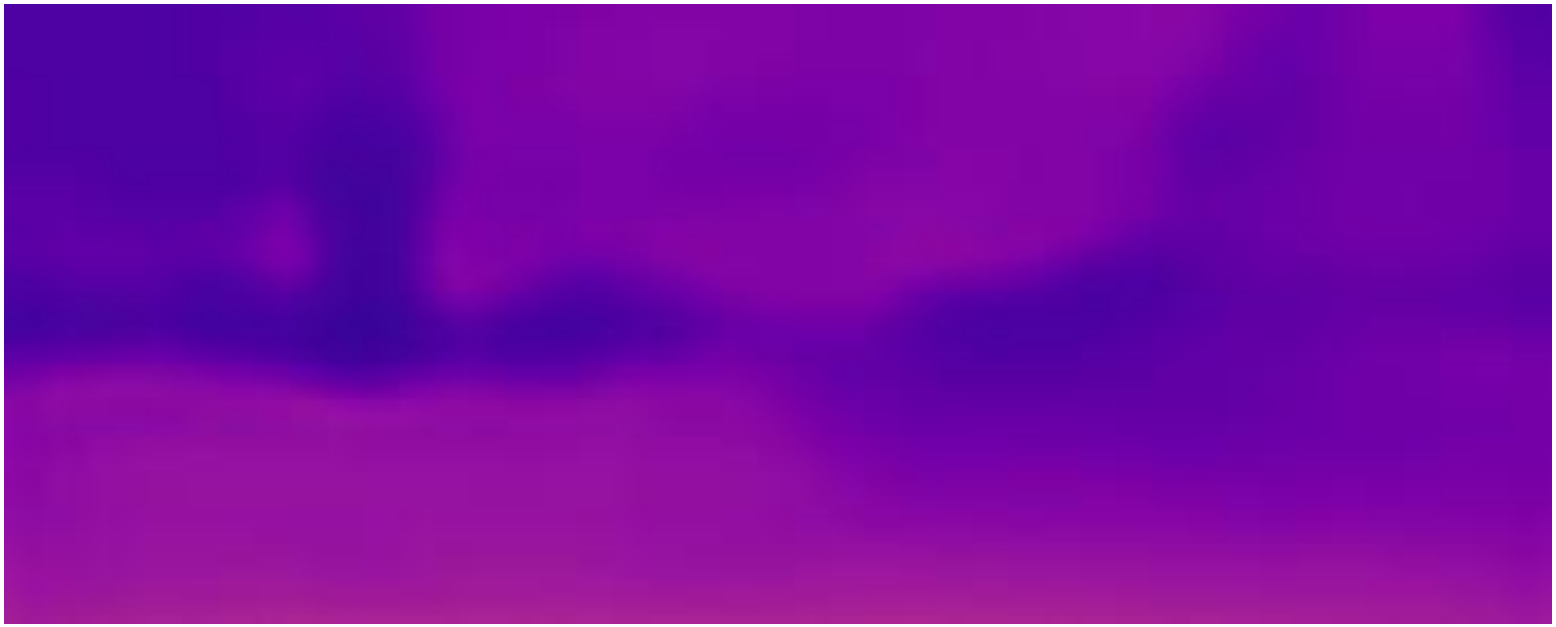
Make3D – Ground truth depth

Make3D – Karsch *et al.* [PAMI 14]

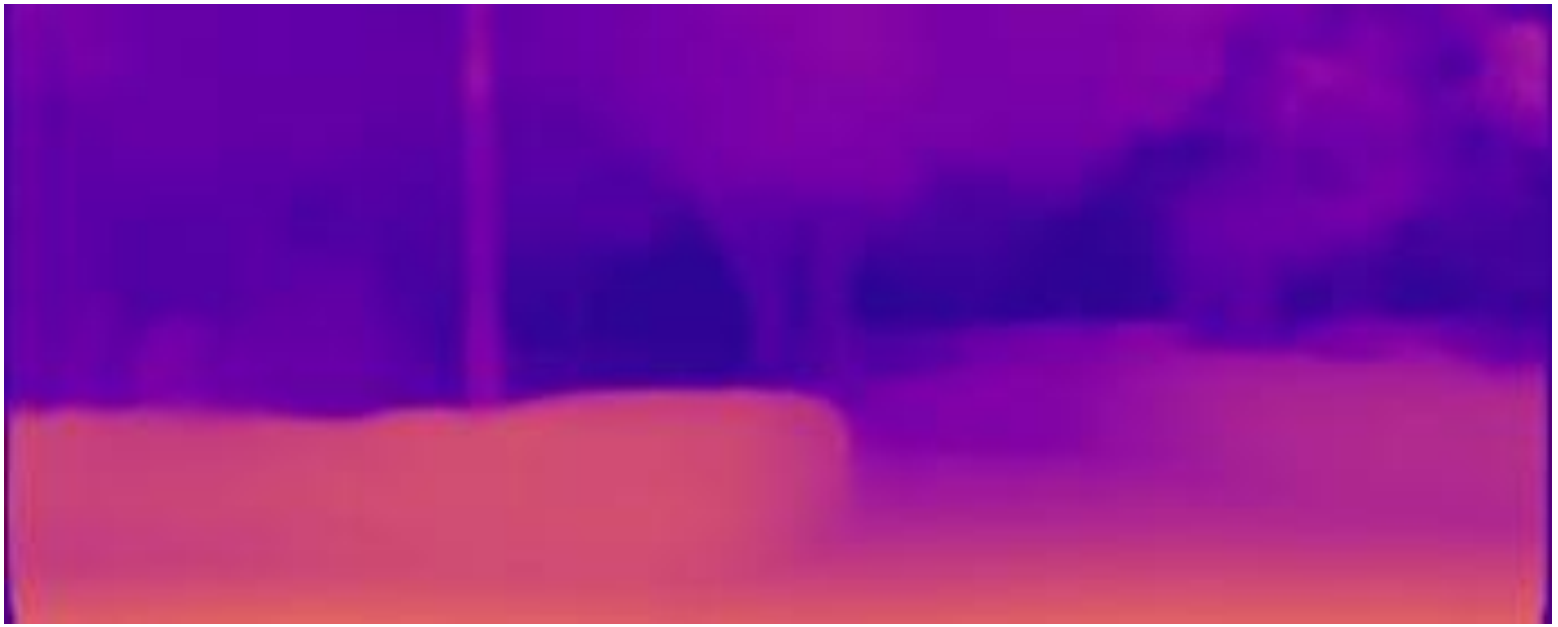


Make3D – Liu *et al.* [CVPR 14]

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]

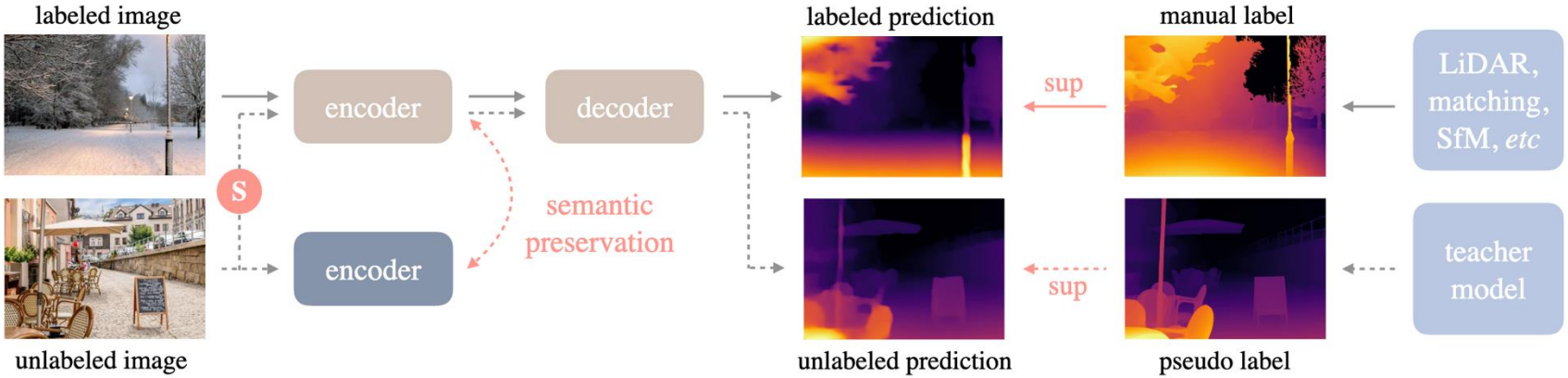
Conclusion

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

Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data. Yang et al. CVPR 2024.

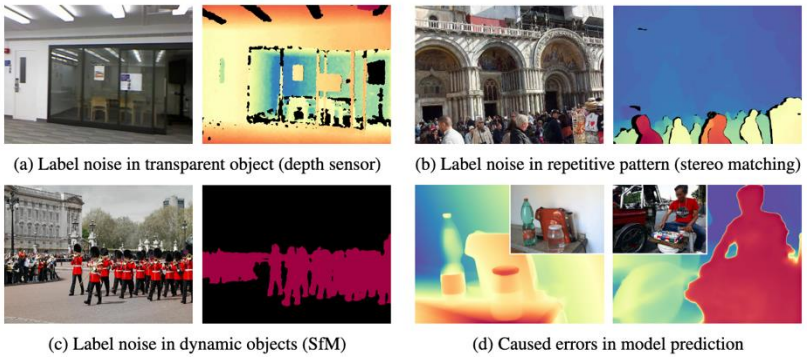
- What is different?
  - Mix of labeled and unlabeled datasets
  - Teacher/student network
  - Encoder with strong semantic prior

Depth Anything



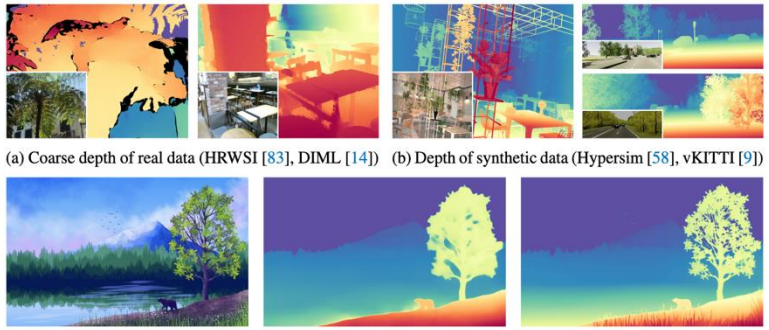
- 1.5M labeled images from multiple datasets
- 65M unlabeled images -> get pseudo ground truth from teacher network
- Train student network from scratch on pseudo ground truth
- Iterate: re-label images with new Teacher, train student
- Introduce strong perturbations (color distortions, color jittering, Gaussian blurring, CutMix)
- Semantic feature alignment based on Dino but with some slack

Depth Anything V2. Yang et al. NeurIPS 2024



(a) Label noise in transparent object (depth sensor) (b) Label noise in repetitive pattern (stereo matching)

(c) Label noise in dynamic objects (SfM) (d) Caused errors in model prediction

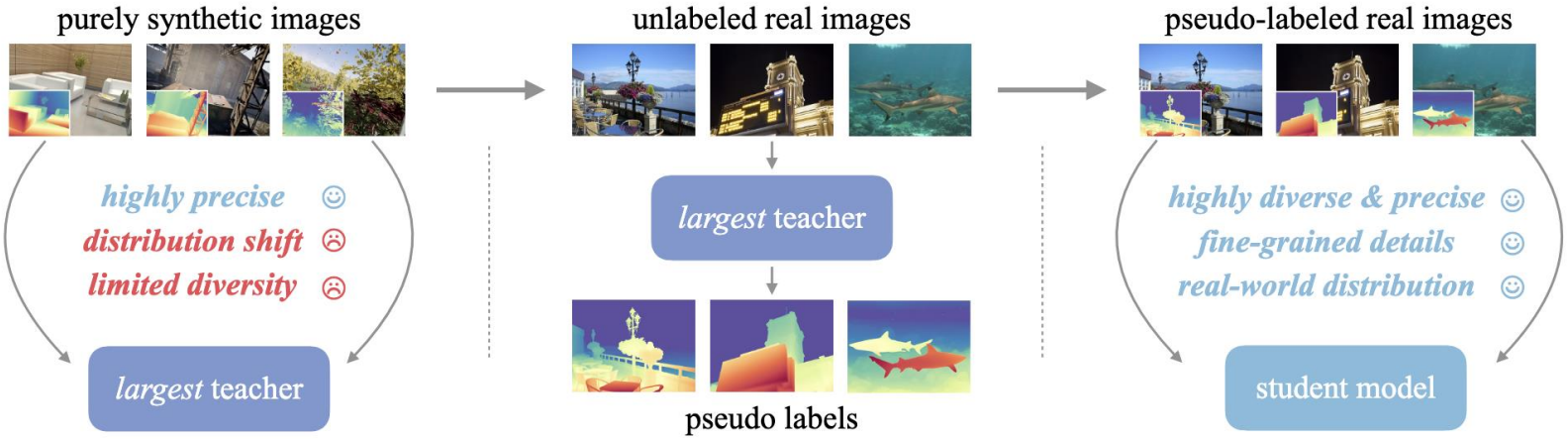


(a) Coarse depth of real data (HRWSI [83], DIML [14]) (b) Depth of synthetic data (Hypersim [58], vKITTI [9])

(c) Predictions of models trained on labeled real images (middle) and synthetic images (right)

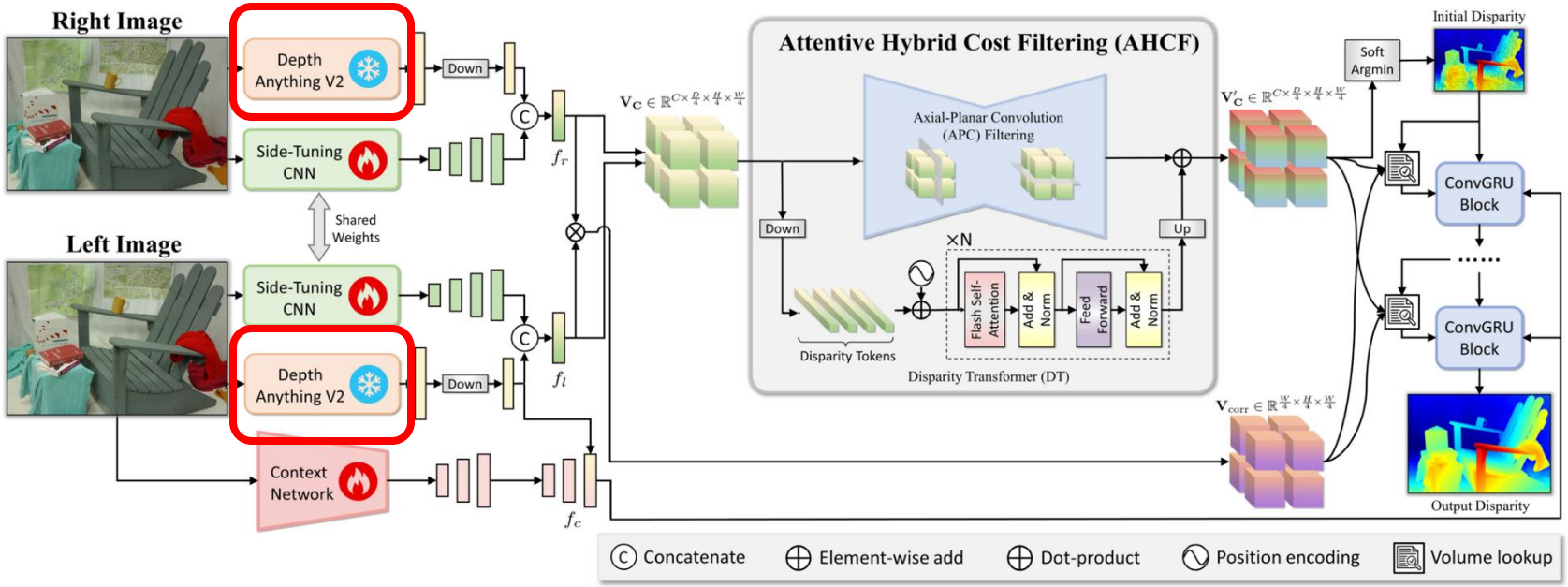
- Teacher trained on synthetic images to mitigate label noise in real datasets
- Student trained on real images to bridge sim2real gap

Depth Anything V2. Yang et al. NeurIPS 2024



- Teacher trained on synthetic images to mitigate label noise in real datasets
- Student trained on real images to bridge sim2real gap

Foundation Stereo

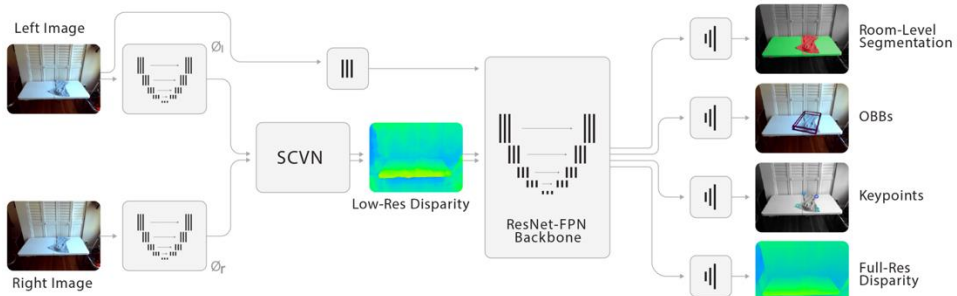


$$\mathbf{V}_{\text{gwc}}(g,d,h,w) = \left\langle \widehat{f}^{(4)}_{l,g}(h,w), \widehat{f}^{(4)}_{r,g}(h,w-d) \right\rangle,$$

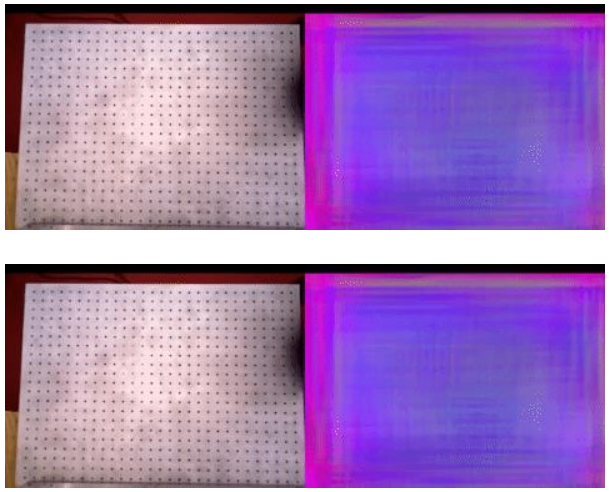
$$\mathbf{V}_{\text{cat}}(d,h,w) = \left[\text{Conv}(f^{(4)}_{l})(h,w), \text{Conv}(f^{(4)}_{r})(h,w-d) \right]$$

$$\mathbf{V}_{\mathbf{C}}(d,h,w) = [\mathbf{V}_{\text{gwc}}(d,h,w), \mathbf{V}_{\text{cat}}(d,h,w)] \qquad (1)$$

Let's use representation learning!



Depth Estimation from Stereo
Supervised Learning



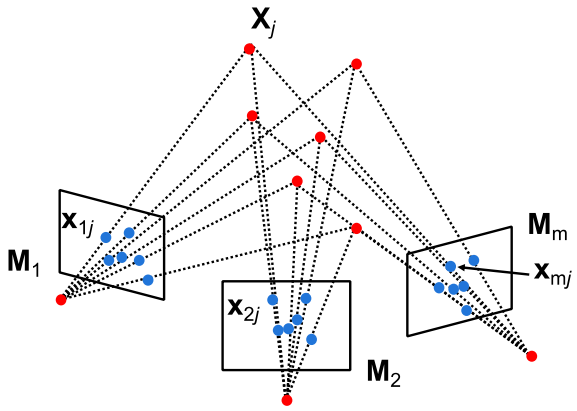
Monocular Depth Estimation
Unsupervised Learning

Image by Yunuk Cha.

Finding Correspondences across Frames
Self-Supervised Learning

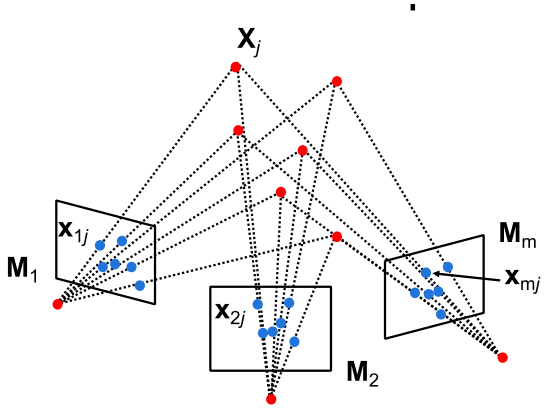
Feature Tracking

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

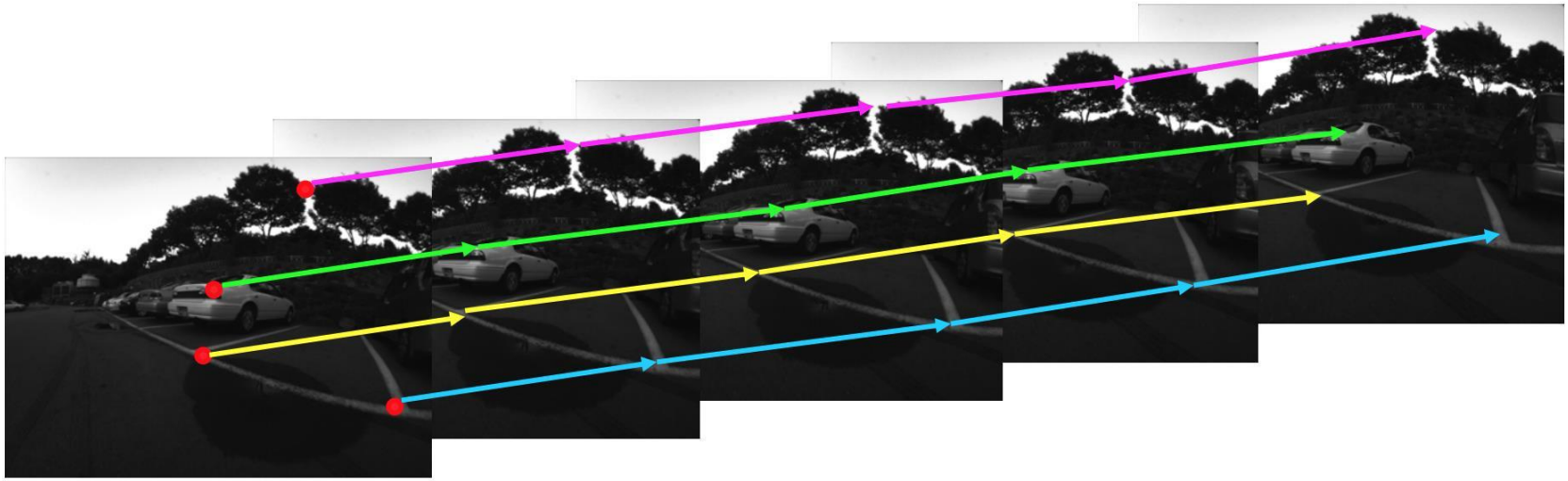
Feature point detection

Slide credit: Yonsei Univ.

Slides Adapted from CS131a.

Problem statement

Feature point tracking



Slide credit: Yonsei Univ.

Slides Adapted from CS131a.

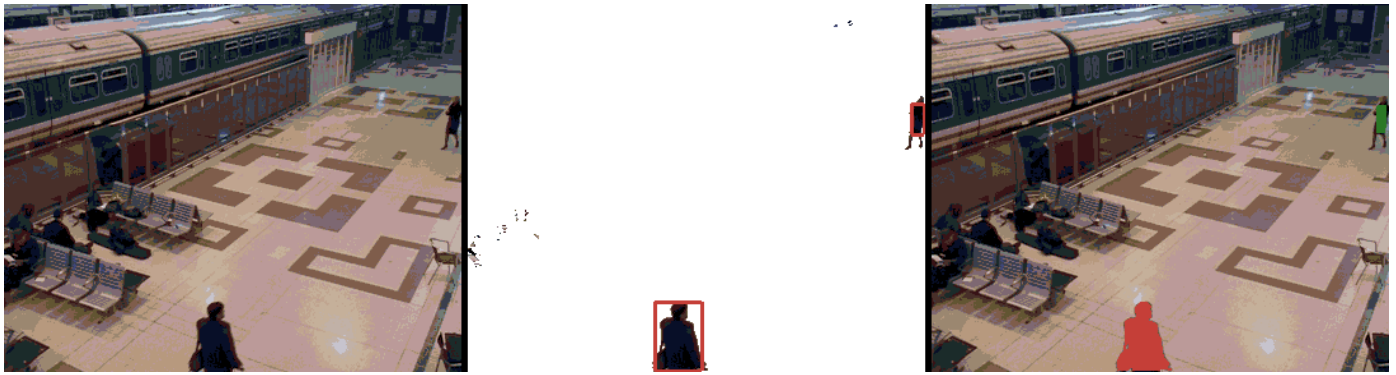
Single object tracking

Slides Adapted from CS131a.

Multiple object tracking

Slides Adapted from CS131a.

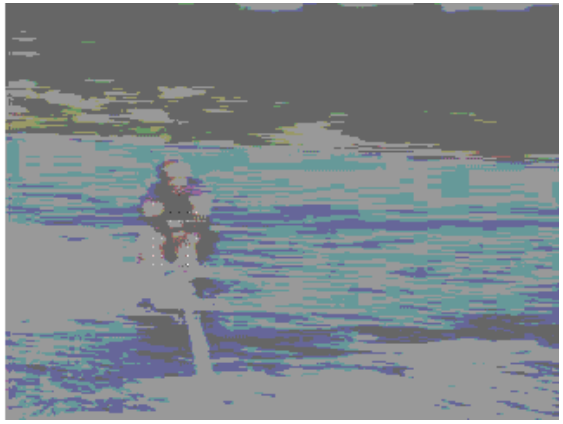
Tracking with a fixed camera



Slides Adapted from CS131a.

Tracking with a moving camera





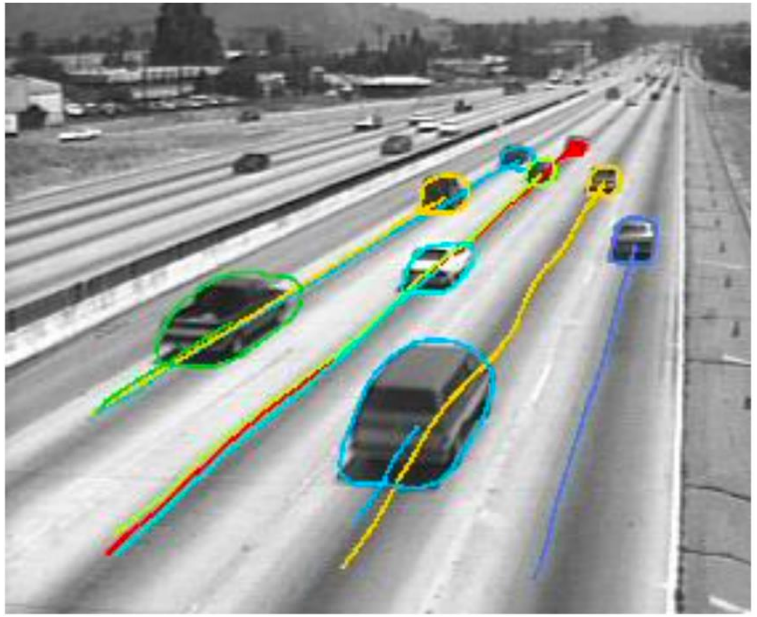
Slides Adapted from CS131a.

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?

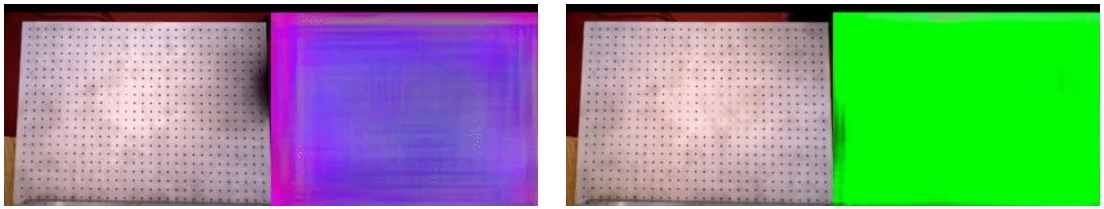
- Regions we can track easily and consistently
- Once we have the good features, we can use simple tracking methods
- Next Lecture: Optical Flow



Dense Object Nets

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

Peter R. Florence, Lucas Manuelli, Russ Tedrake

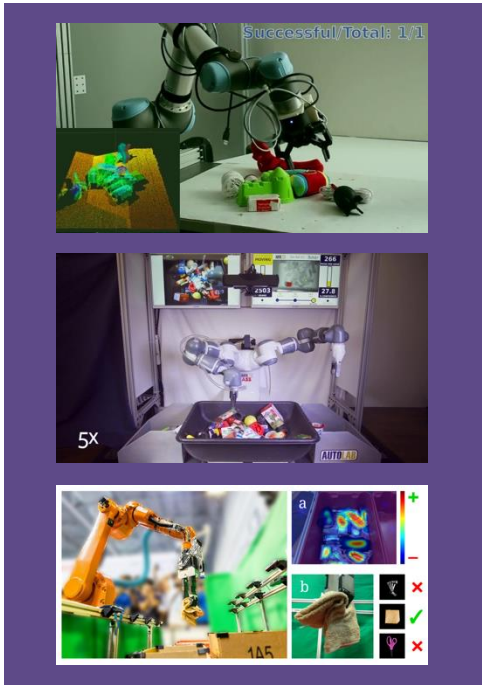


Slides adapted from CS326 by Kevin Zakka and Sriram Somasundaram

Motivation

RL

task-specific



Grasp feature-learning

no task-specificity

Grasp segmentation

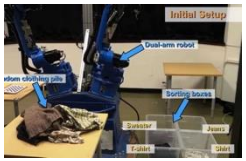
coarse

no task-specificity

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

Wish List

Deformable



Task agnostic

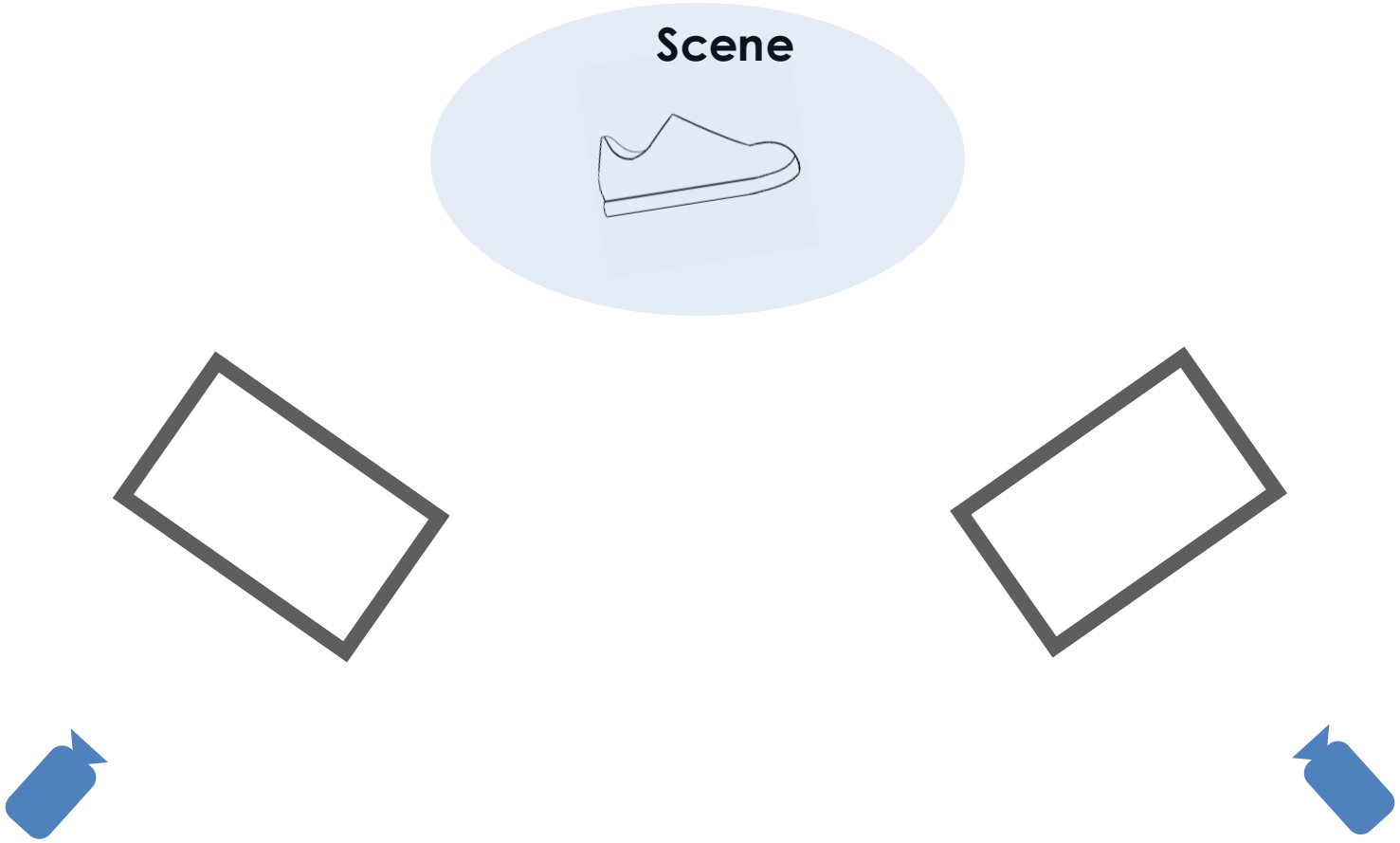
Self-supervised

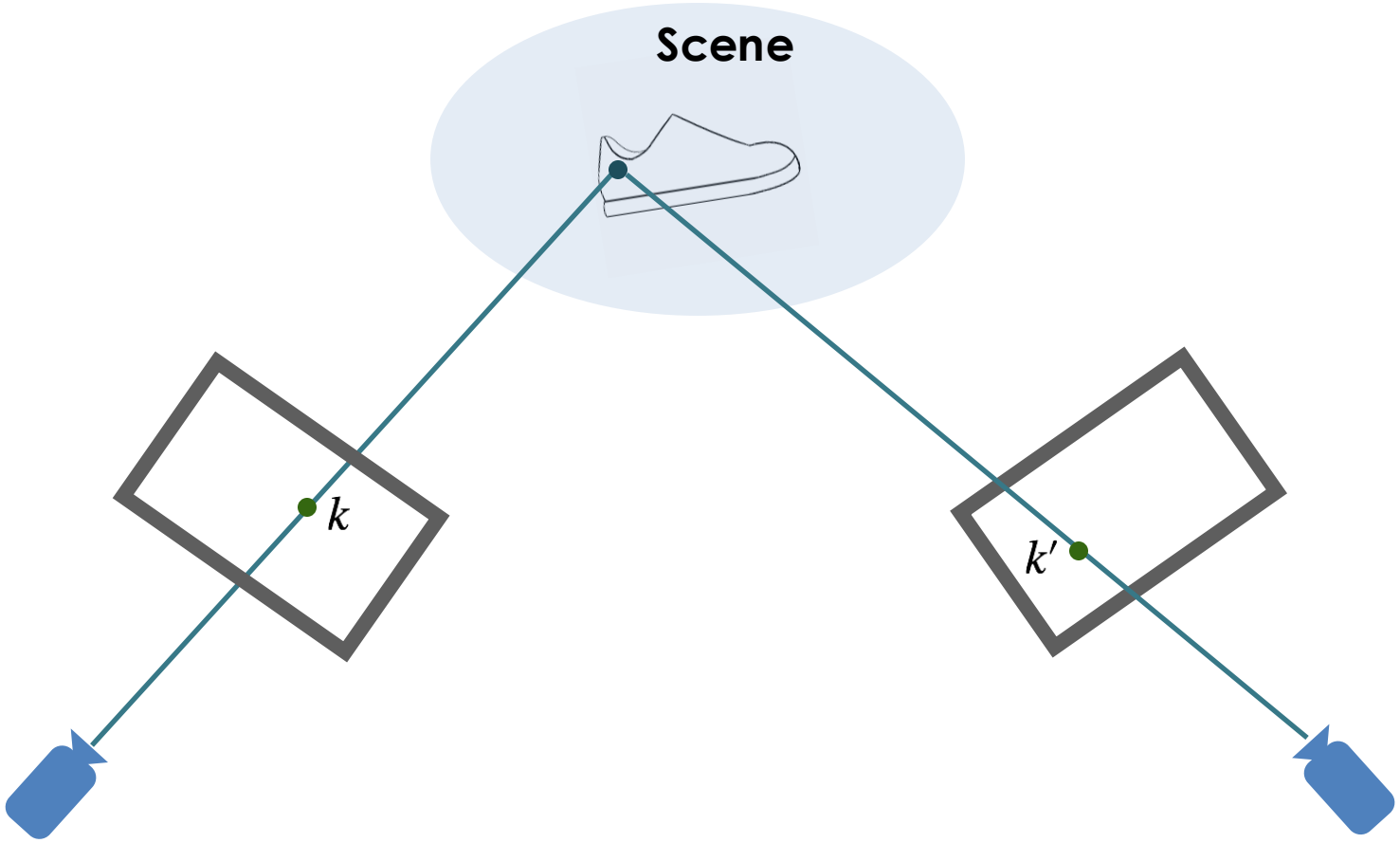
3D perception

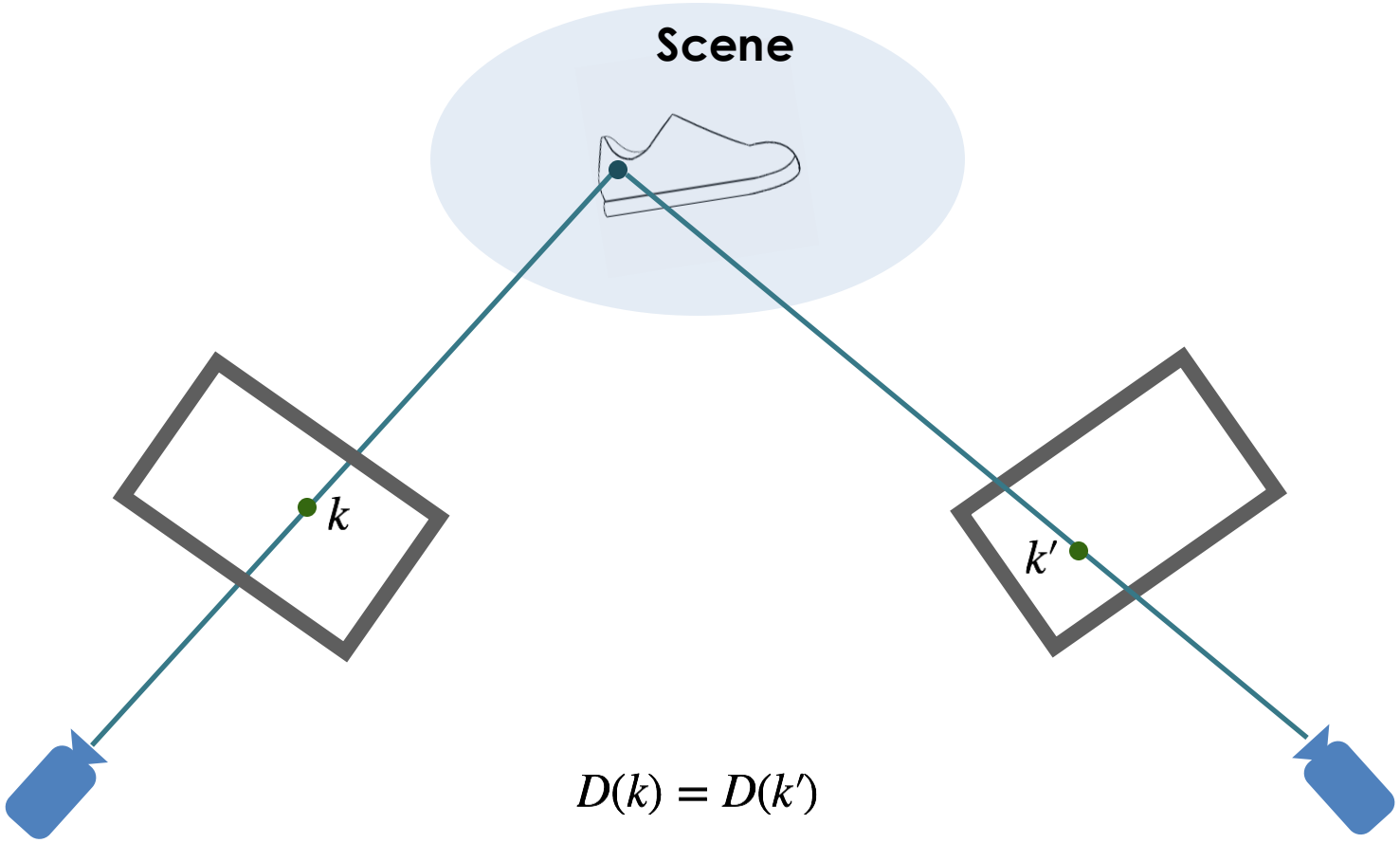
Some Representations

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′)

Feature detector

Feature descriptor

$I \rightarrow \{I[x_1, y_1], I[x_2, y_2], \dots\}$

Area around pixel $k \rightarrow D(k)$

Area around pixel k → $D(k)$

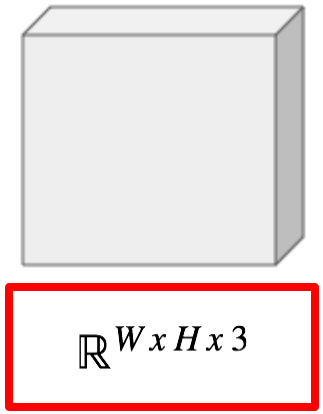
Area around pixel k → $D(k)$ Area around pixel k → $D(k)$

Features descriptors should be invariant under transformation

Paper Overview

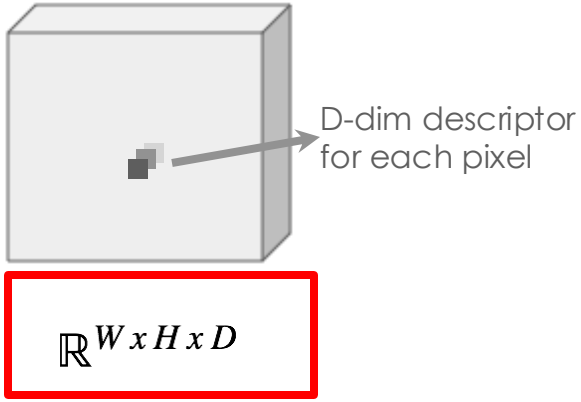
Dense Descriptors

Input is an RGB image



$f(\cdot)$

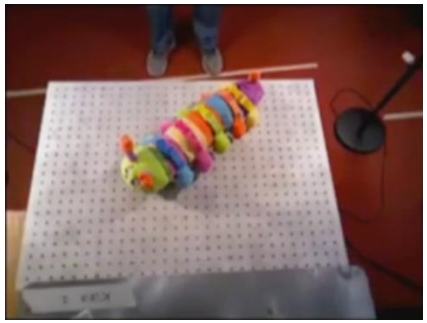
Output



Pay attention to the difference in Dimensionality

Dense Descriptors

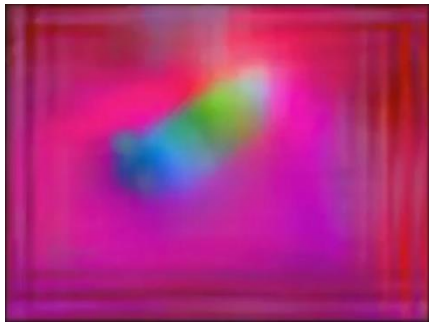
Input is an RGB image



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

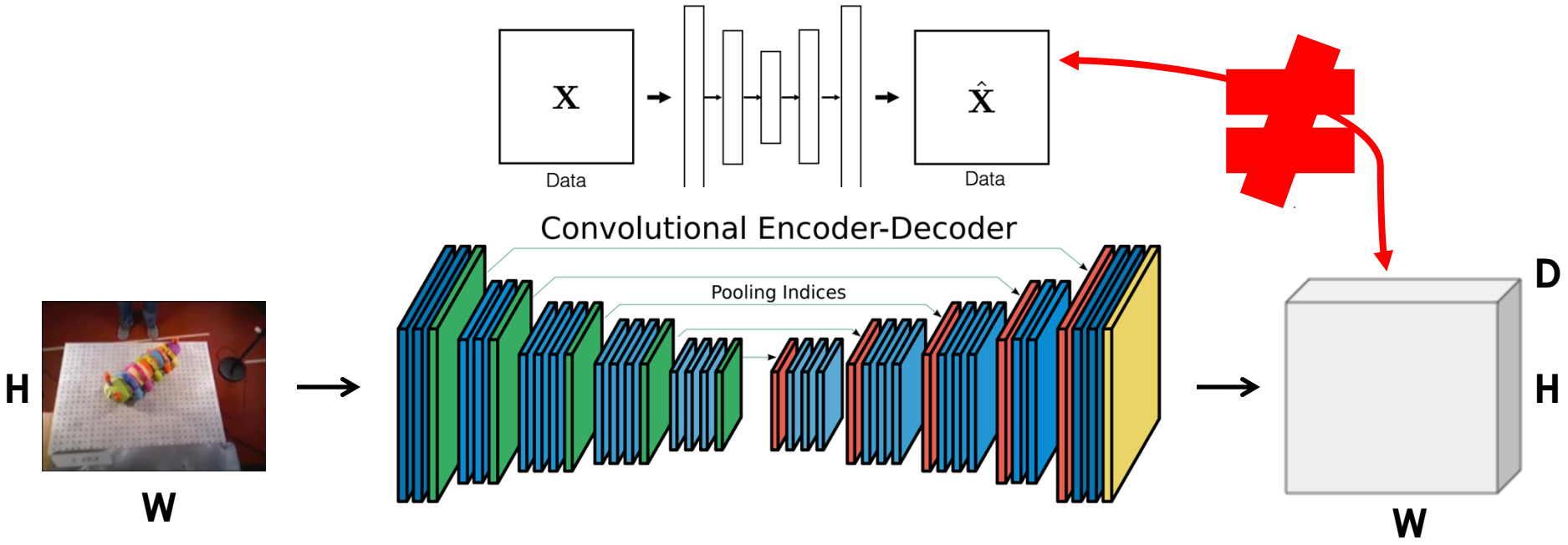
$f(\cdot)$

Output

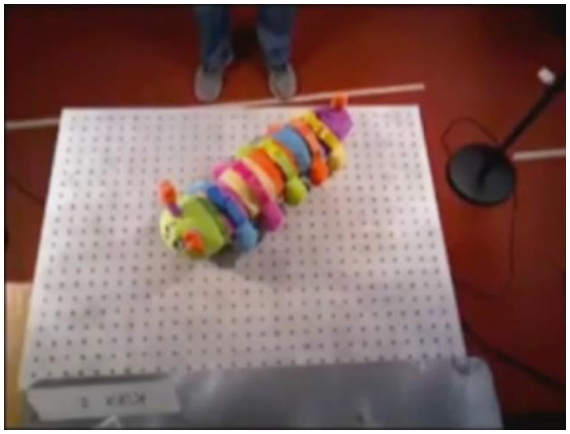


$\mathbb{R}^{W \times H \times D}$

Network Architecture



Pixelwise Contrastive Loss

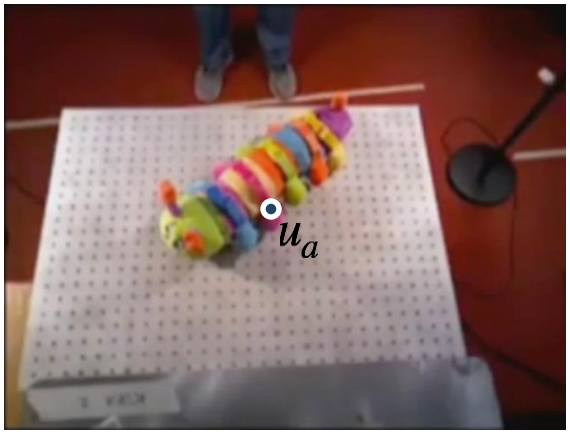


I_a

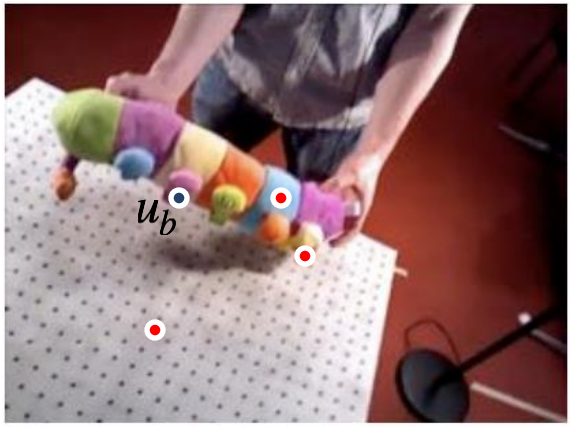
I_b

Hadsell et al., CVPR 2006

Pixelwise Contrastive Loss



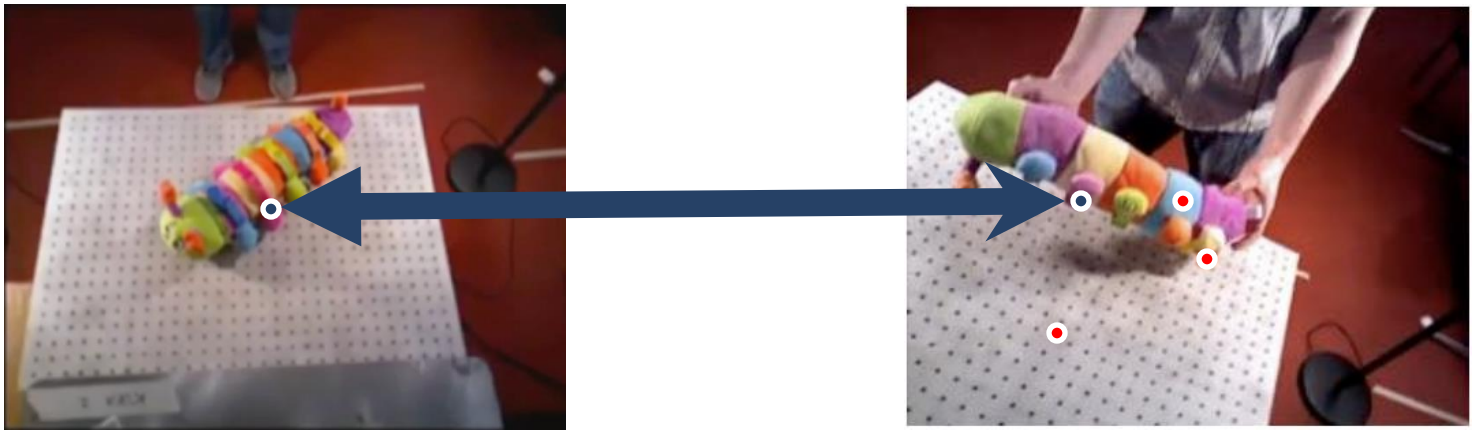
I_a



I_b

Assumption: Ground truth Correspondences Given

Pixelwise Contrastive Loss - Matches

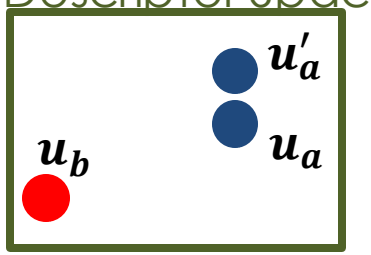


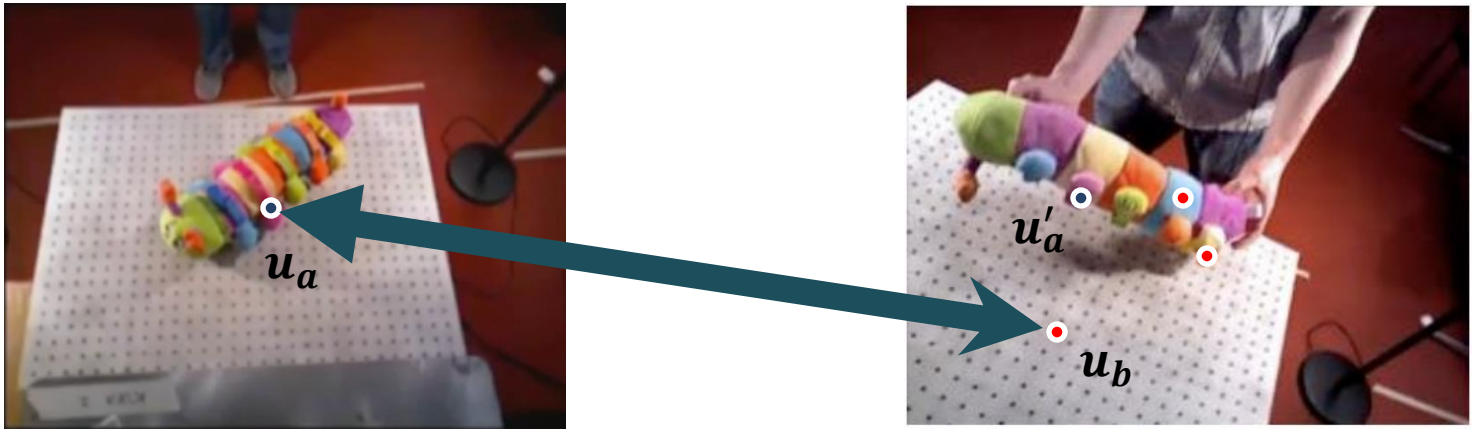
$$L_{\text{matches}}(I_a, I_b) = \frac{1}{N_{\text{matches}}} \sum_{N_{\text{matches}}} D(I_a, u_a, I_b, u_b)^2$$

Distance in Descriptor Space

Pixelwise Contrastive Loss – Non-Matches

- Training time





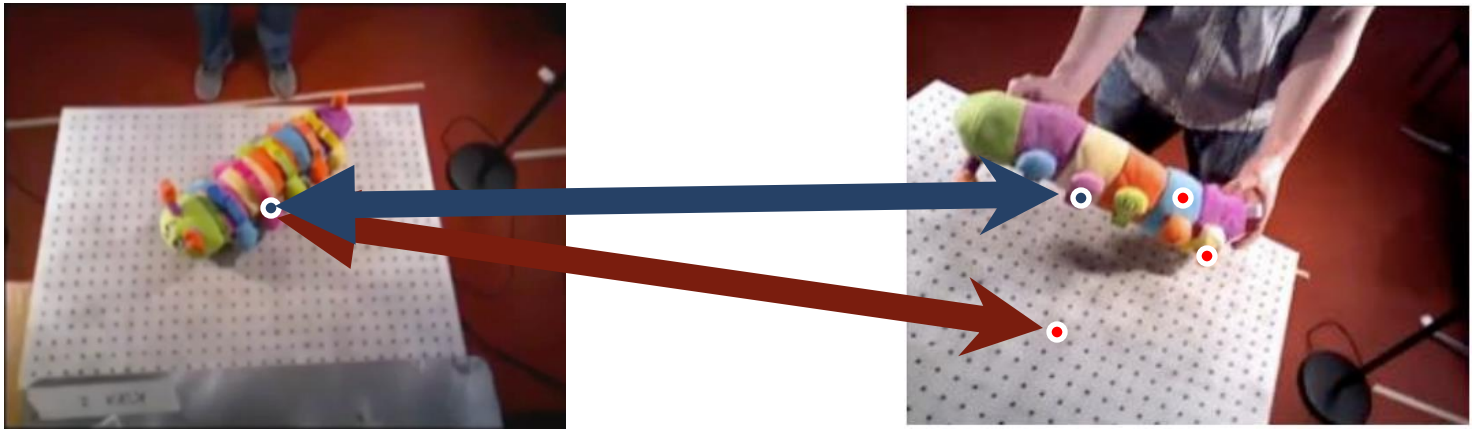
Max distance

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

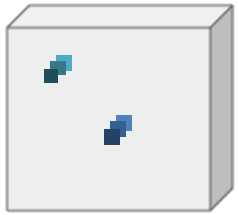
Distance in Descriptor Space

You want this to be large for non-matches

Pixelwise Contrastive Loss



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

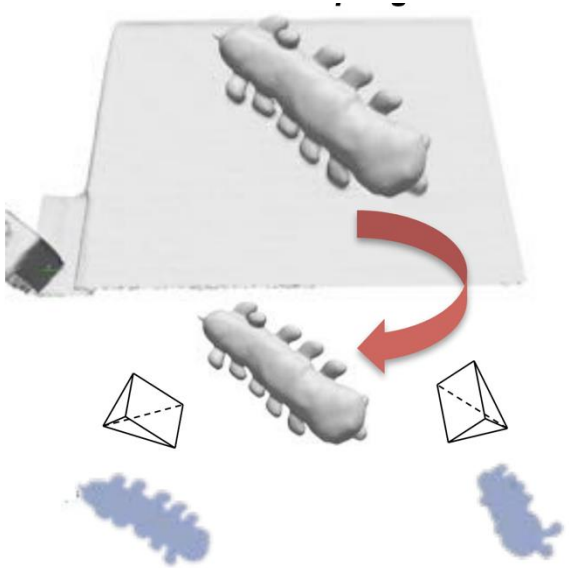


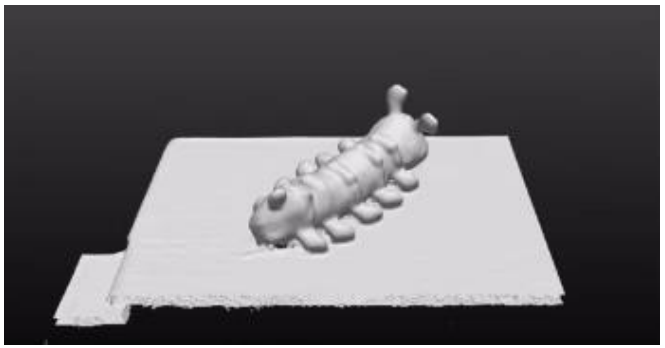
$\mathbb{R}^{W \times H \times D}$

Loss: Reconstruction + Contrastive Loss

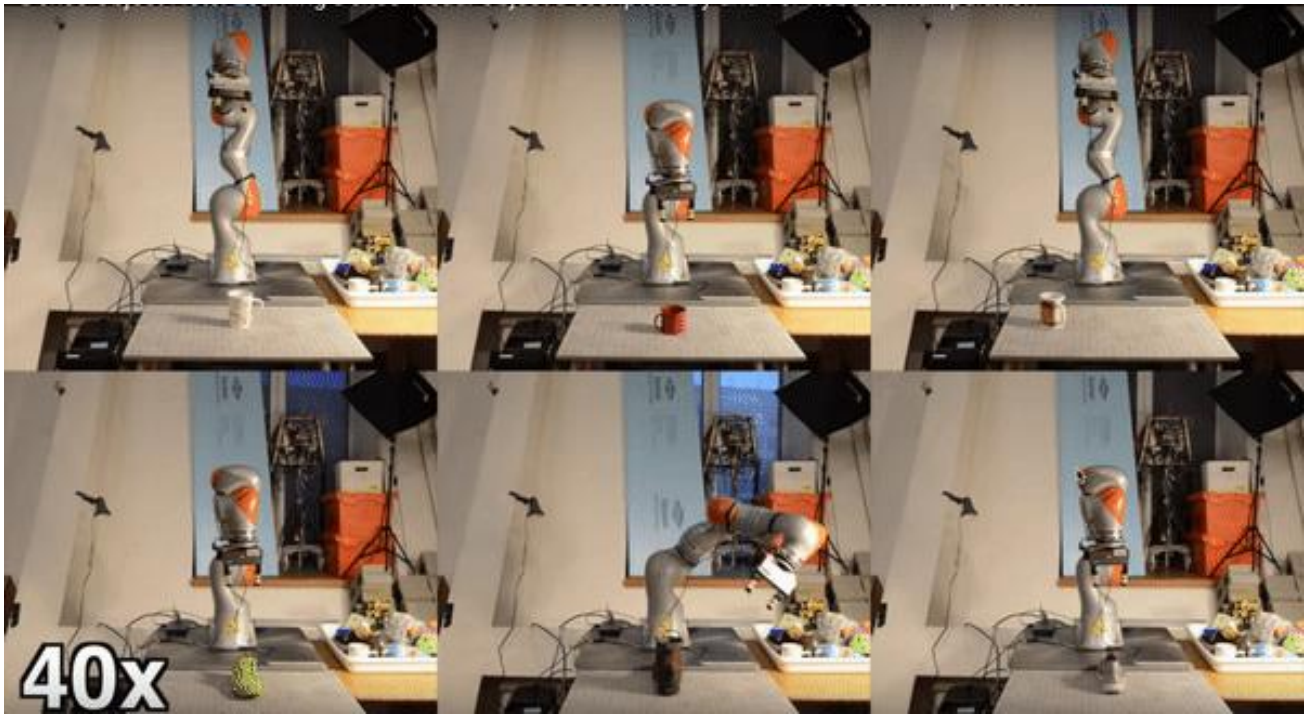
How can we generate these ground-truth correspondences?

3D Reconstruction based Change Detection and Masked Sampling

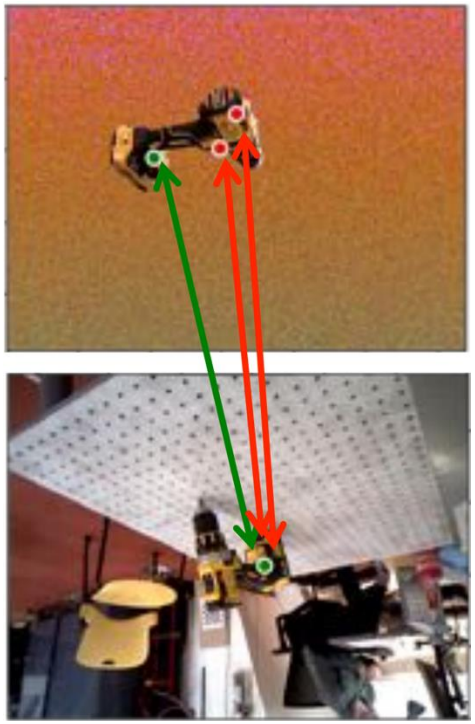




Autonomous and Self-supervised Data Collection



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

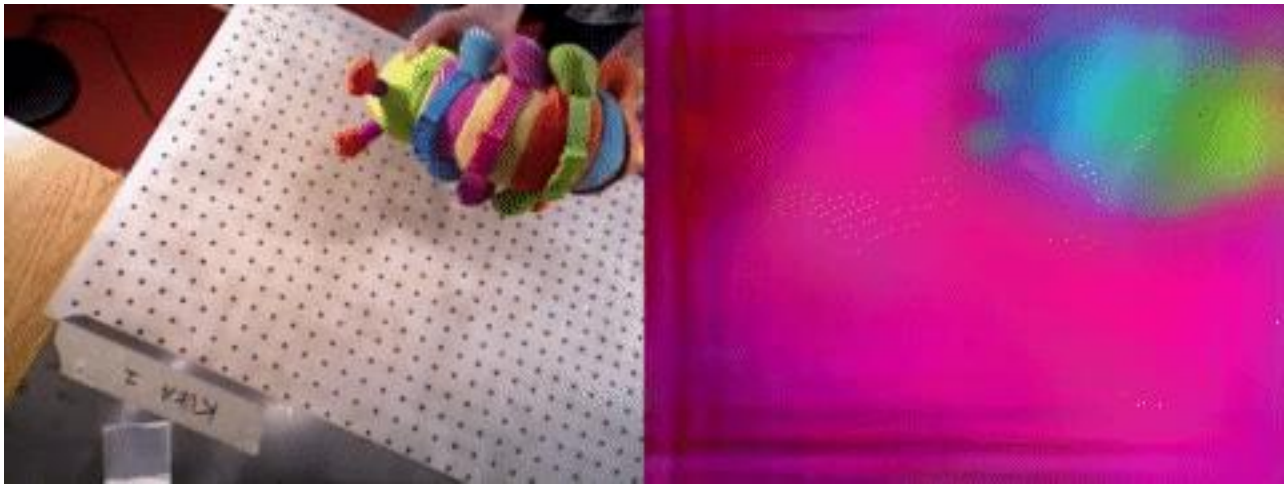
Results

Experiments

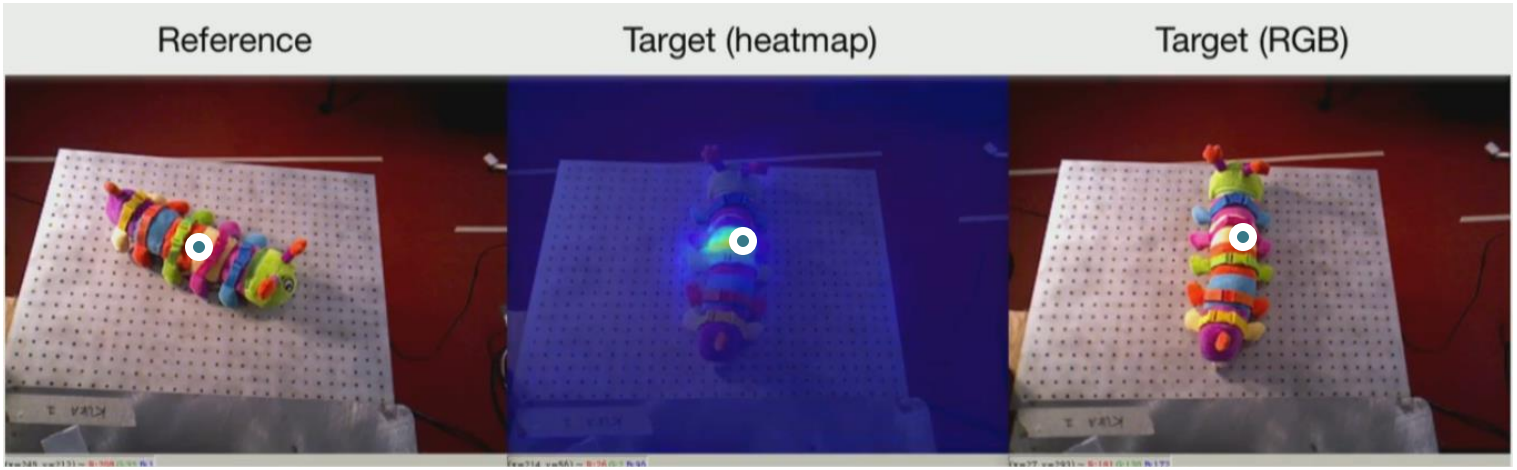
1) Can we acquire descriptors that can generalize across classes of objects and can be distinct for each object instance?

2) Can we apply dense descriptors to robotic manipulation?

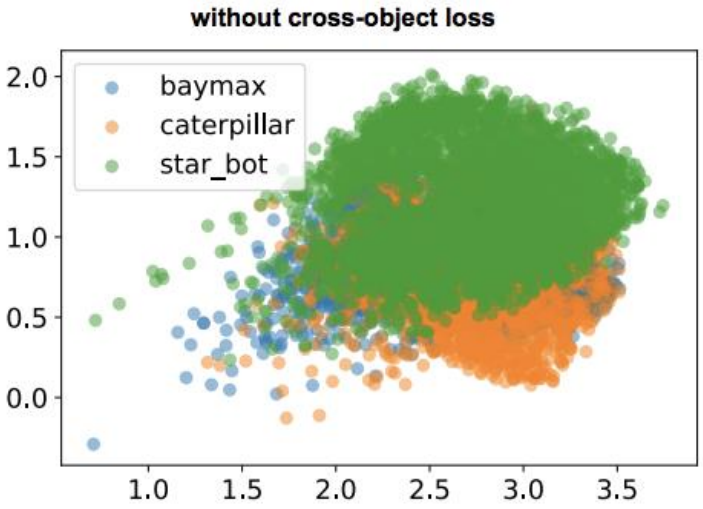
Single Object

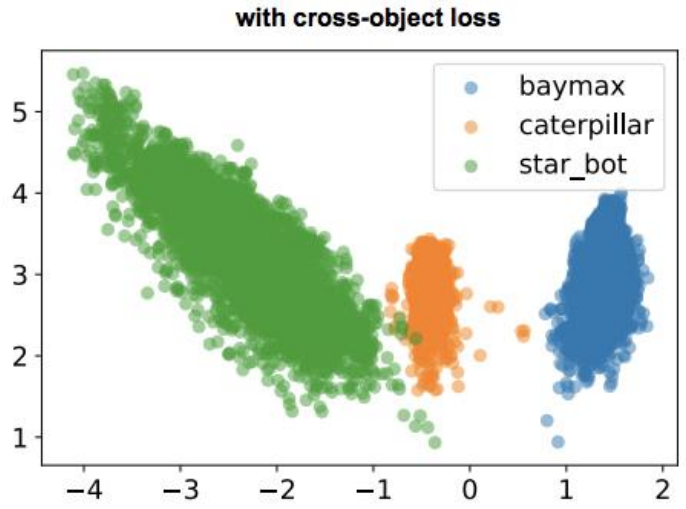


Learned Dense Correspondences

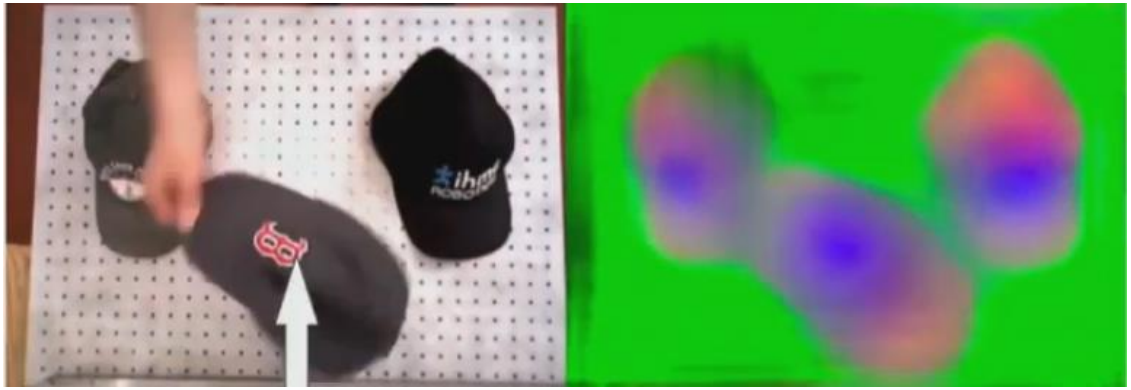


Multi-Object Unique Descriptors





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

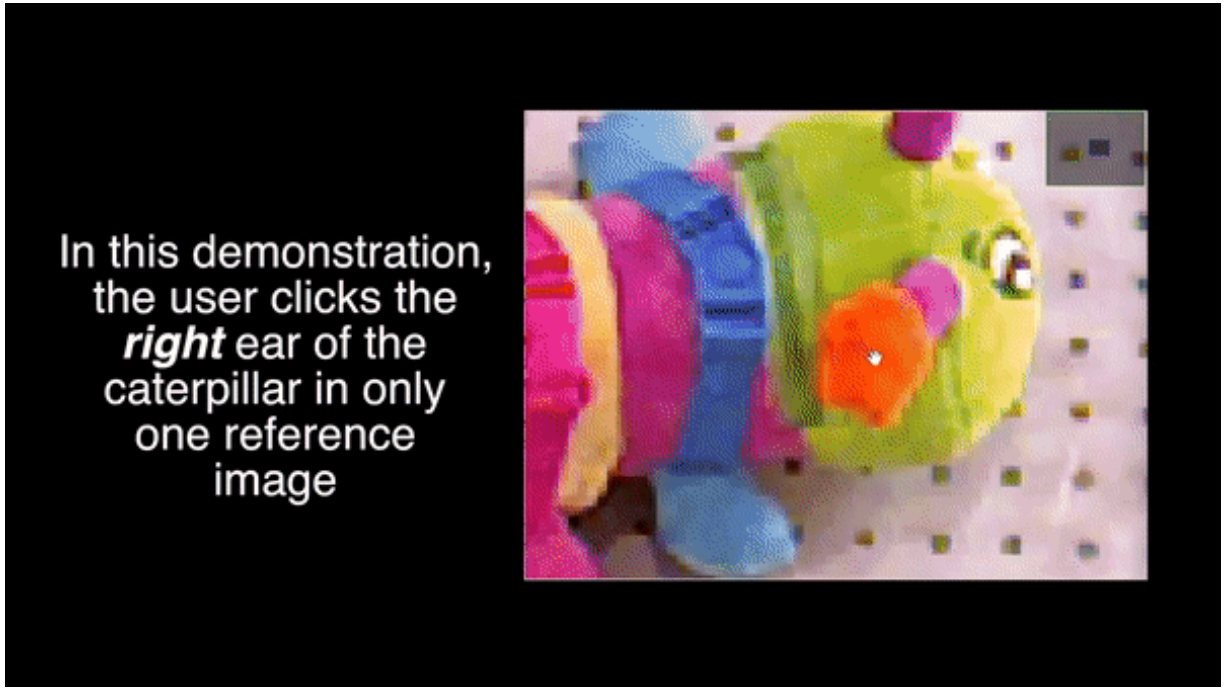
2) Can we apply dense descriptors to robotic manipulation?

Results: Robotics Showcase

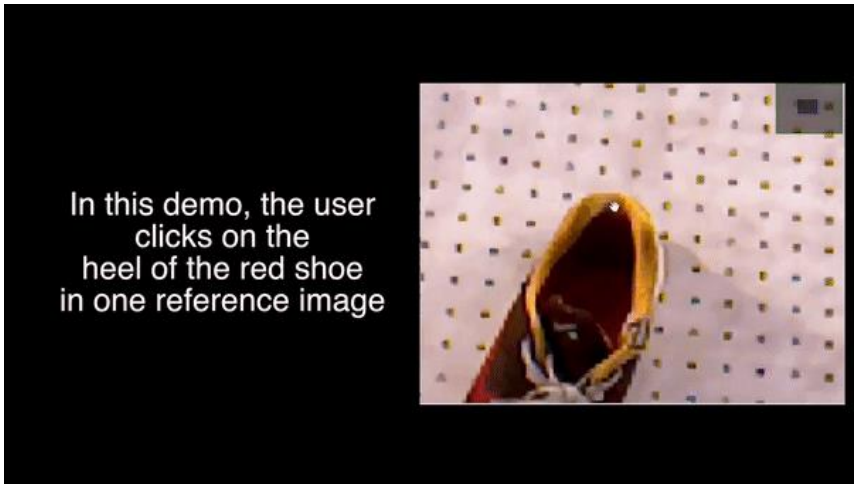
Goal: Perform grasps on target object with a real robot based on selected grasp points on a reference object

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

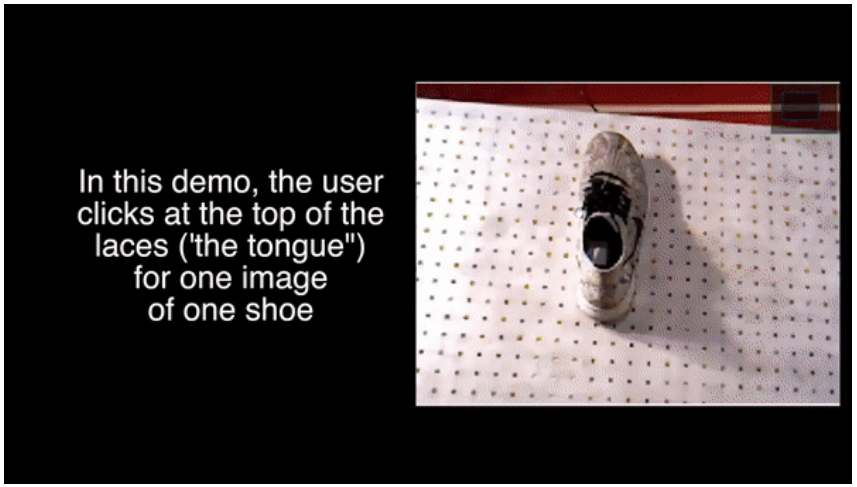
Dense Correspondence for Manipulation

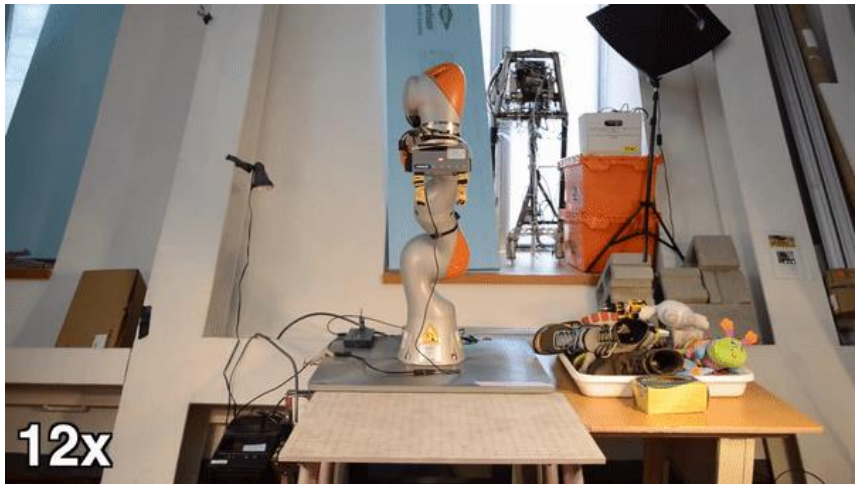


Instance Specific Manipulation



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)

Emerging Properties in Self-Supervised Vision Transformers. Caron et al. ICCV. 2021

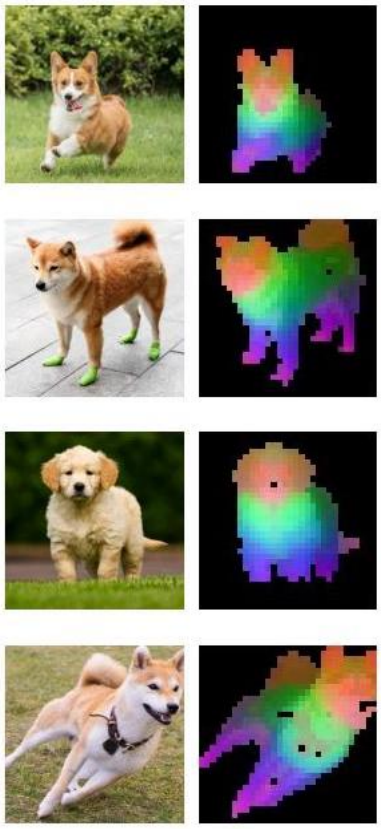


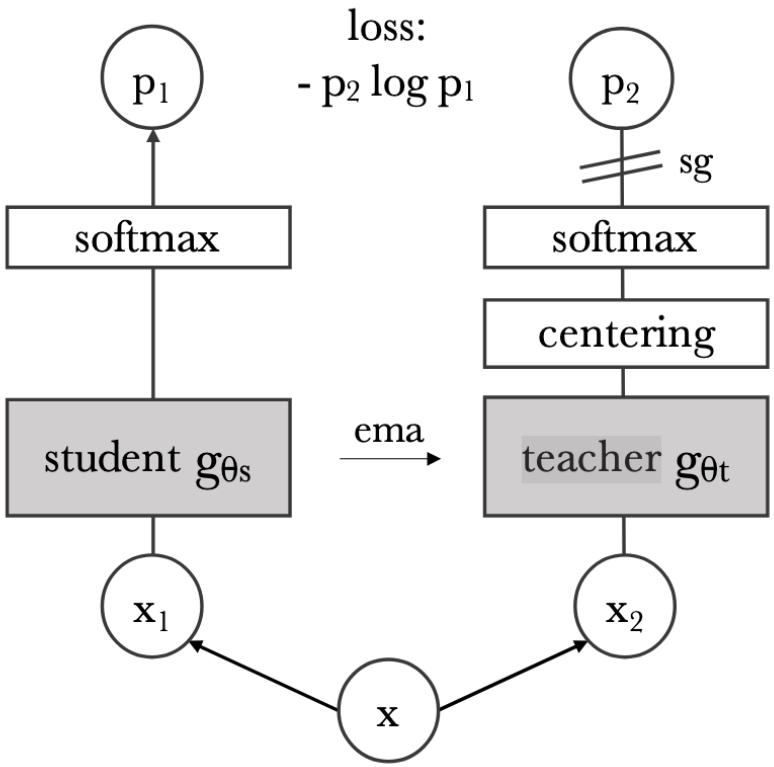
Image by Yunuk Cha.

- Learns local features that are consistent across time and semantic classes
- Student/Teacher Architecture
- Uses Vision Transformer

Background: Vision Transformer

- Image divided into Patches; patches are vectorized
- Fed into transformer with positional encoding

Emerging Properties in Self-Supervised Vision Transformers. Caron et al. ICCV. 2021

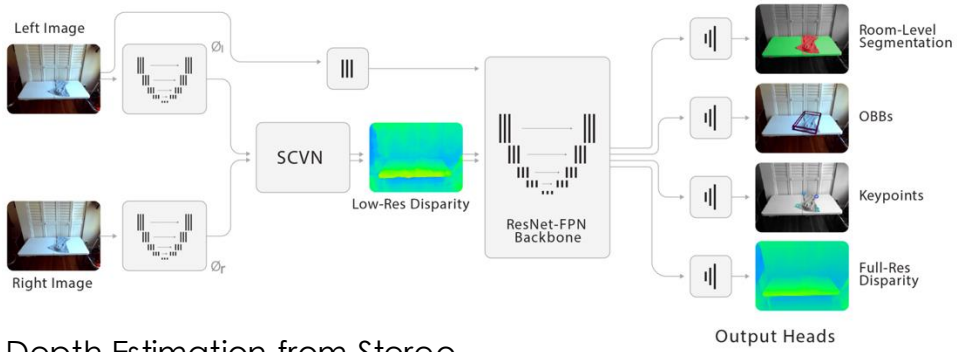


- Train student to match output of teacher
- Input: Images (here only 2)
- Output: Distribution over image categories
- Loss: Cross Entropy loss
- Teacher update with Exponential Moving Average (EMA) over past iterates of student

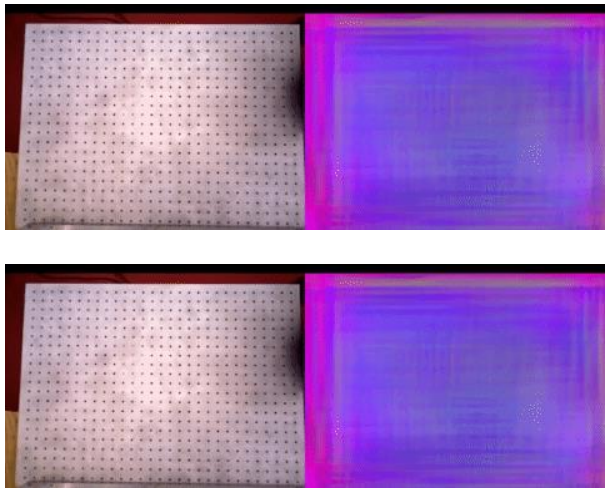
Emerging Properties in Self-Supervised Vision Transformers. Caron et al. ICCV. 2021

- Local and global views extracted from images
  - crops have same class label as original image
- Teacher gets only global view
- Student gets both, global and local views

Let's use representation learning!



Depth Estimation from Stereo
Supervised Learning

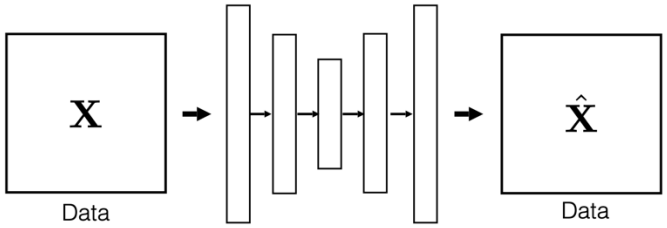


Monocular Depth Estimation
Unsupervised Learning

Image by Yunuk Cha.

Finding Correspondences across Frames
Self-Supervised Learning

Summary



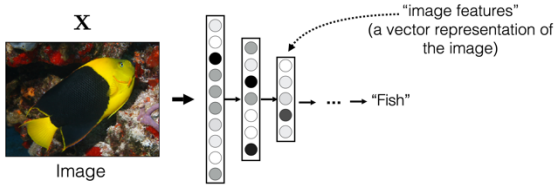
- Hourglass structure for representation learning
- Mapping either to depth or to descriptors
  - Descriptors are used for localization, robot grasping, tracking
  - Keypoint matching
- Training is supervised, unsupervised, self-supervised by exploiting structure that you know about the problem
  - Eases demand for ground truth labels
- Known structure can be used to generate training data (ground truth depth, correspondences, optical and scene flow, semantics, …)

Common themes in newer models

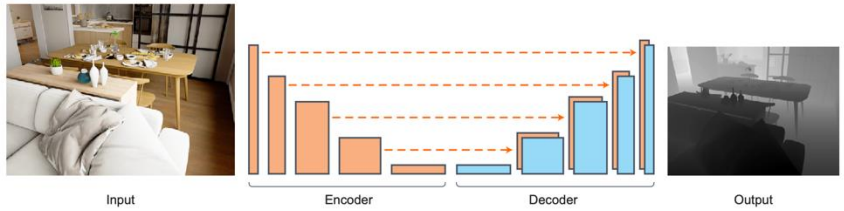
- Cost volume for depth estimation
- Implicit learning of correspondences
- Simulated training data
- Student / Teacher distillation
- Vision transformer

Learning Goals for Upcoming Lectures

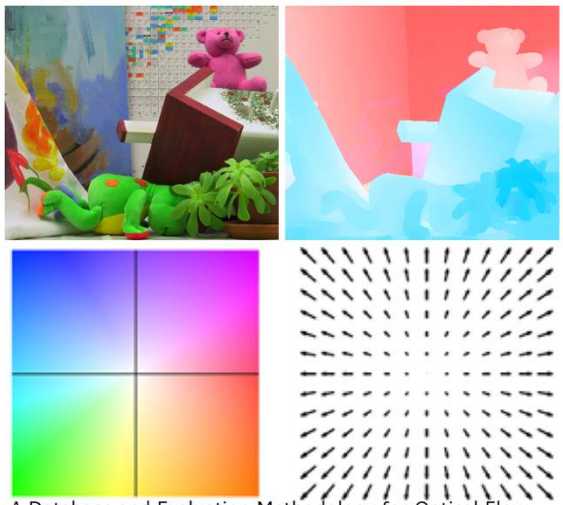
Representations & Representation Learning



Using Representation Learning for Depth Estimation and Finding Correspondences

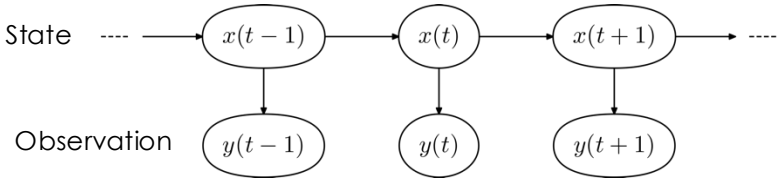


Optical & Scene Flow

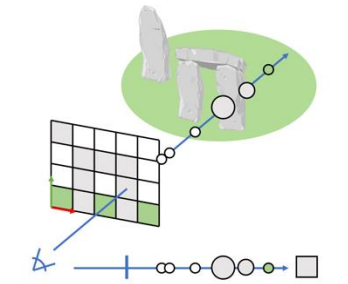


A Database and Evaluation Methodology for Optical Flow.
Baker et al. IJCV. 2011

Optimal Estimation



Neural Radiance Fields



CS231

Introduction to Computer Vision

Next Lecture:

Optical Flow and Scene Flow