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

Optimal Estimation

Neural Radiance Fields

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

Input images → "image features" (a vector representation of the image) → Intermediate representation

Training data

\{x_1, y_1\}
\{x_2, y_2\}
\{x_3, y_3\}
\ldots

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

Loss function/Cost

Classifier

\[ f : X \rightarrow Y \]
Learning without Labels

Training data

\{x_1, y_1\}
\{x_2, y_2\}
\{x_3, y_3\}
\ldots

\rightarrow

f : X \rightarrow Y

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

Given an input image $X$, the autoencoder process involves:

1. Encoding the input image into an intermediate representation $z$.
2. Reconstrucing the image $\hat{X} = \mathcal{G}(X)$ using the decoder $\mathcal{G}$.

The goal is to minimize the reconstruction loss by finding the optimal $\mathcal{G}$.

Mathematically, this is expressed as:

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

**Reconstruction loss to minimize by finding optimal $\mathcal{G}$**
Data Compression & Task Transfer

Diagram:
- Encoder
- Decoder
- Classifier
- Intermediate repr.
  e.g., image classification
Self-Supervision

\[
G(X) = \hat{X} \\
G(X_1) = \hat{X}_2
\]
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 ground truth
- Image Synthesis Error for right image of a stereo pair given left image
- 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 self-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
Let’s use representation learning!

Depth Estimation from Stereo Supervised Learning

Finding Correspondences across Frames Self-Supervised Learning

Monocular Depth Estimation Unsupervised Learning
Epipolar Constraint (Lecture 6)

\[ p^T \cdot F \cdot p' = 0 \]

- \( l = F \cdot p' \) is the epipolar line associated with \( p' \)
- \( l' = F^T \cdot p \) is the epipolar line associated with \( p \)
- \( F \cdot e' = 0 \) and \( F^T \cdot e = 0 \)
- \( F \) is a 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

![Diagram of point triangulation](image)

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

\[
\text{disparity} = p_u - p'_u \propto \frac{B \cdot f}{z}
\]

Disparity is inversely proportional to depth \( z \)!
Disparity maps

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

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

[Eq. 1]

Stereo pair

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

$\bar{u}' = \bar{u} + d = \bar{u} - 1$
The Correspondence Problem

Occlusions

- Occluded pixel
- Good match

Homogenous regions

- Hard to match pixels in these regions

Repetitive Patterns

- Red boxes indicating repetitive patterns
Can we learn a similarity function to find corresponding points?

Supervised Learning of Disparity map.

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?
An image is primarily composed of:
1. Geometry
2. Appearance - 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.

\[ c(x_1, x_2) = \sum_{o \in [-k,k] \times [-k,k]} \langle f_1(x_1 + o), f_2(x_2 + o) \rangle \]

- \( f_1 \) = Feature
- \( x_1 \) = Image Patch location
- \( k \) = ½ Image Patch dimension
Randomize over Geometry
Densely Annotated Scenes
3D Scene Understanding
Let’s use representation learning!

Depth Estimation from Stereo Supervised Learning

Monocular Depth Estimation Unsupervised Learning

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?
Unsupervised Monocular Depth Estimation with Left-Right Consistency

Clément Godard¹ Oisin Mac Aodha² Gabriel J. Brostow¹

¹University College London ²Caltech
Result
How do we usually get depth?
Why monocular?
Previous approaches - Supervised

Input color → Model → Output depth → Target depth

Loss
Previous approaches - Supervised

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

Loss

New!
KITTI 2015
IR Structured Light

• Does not work well outside
Depth from stereo

\[ \text{Disparity} \propto \frac{1}{\text{Depth}} \]
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

Input colors → CNN → Output disparity → Sampler → Output color → Target color

Loss

Lecture 11
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
- \( \alpha \) = weight

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

Luminance \hspace{1cm} Contrast \hspace{1cm} Structure

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

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

Complete Loss

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

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

- \( N \) = # of pixels
- \( i, j \) = index of the pixels
- \( d \) = disparity in left/right image
- \( \partial_x d_{i,j} \) = Gradient of disparity
- \( \partial_x I_{i,j} \) = Image Gradient
Unsupervised depth estimation - This method

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

Input colors → CNN → Output disparities → Sampler → Output colors → Target colors

LR Loss → Smoothness Loss → Loss
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
KITTl – 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
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?
    • Kuzniyetsov 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!
Examples in Industry

Robotics Today - A Series of Technical Talks
Watch | Upcoming Seminars | Past Seminars | Organizers | Contact

"Robotic 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 safely 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.

SimNet. Corl '21

Teaching Robot to help People in their home.
Let’s use representation learning!

Depth Estimation from Stereo Supervised Learning

Monocular Depth Estimation Unsupervised Learning

Finding Correspondences across Frames Self-Supervised Learning
Feature Tracking

Structure From Motion Problem

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

- $x_{ij} = M_i X_j$, $i = 1, \ldots, m$, $j = 1, \ldots, n$

From the $m \times n$ observations $x_{ij}$, estimate:

- $m$ projection matrices $M_i$
- $n$ 3D points $X_j$
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

Slides Adapted from CS131a.
Single object tracking

Slides Adapted from CS131a.
Multiple object tracking

Slides Adapted from CS131α.
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

\[ D(k) = D(k') \]
Feature detector

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

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

Feature descriptor
Area around pixel $k \rightarrow D(k)$
Features descriptors should be invariant under transformation

Area around pixel $k \rightarrow D(k)$  
Area around pixel $k \rightarrow D(k)$
Paper Overview
Dense Descriptors

Input is an RGB image

Output

\[ \mathbb{R}^{W \times H \times 3} \] \[ \rightarrow \] \[ f(\cdot) \] \[ \mathbb{R}^{W \times H \times D} \]

Pay attention to the difference in Dimensionality
Dense Descriptors

Input is an RGB image

\( \mathbb{R}^{W \times H \times 3} \)

Output

\( \mathbb{R}^{W \times H \times D} \)

\( f(\cdot) \)
Network Architecture
Pixelwise Contrastive Loss

$I_a$

$I_b$

Hadsell et al., CVPR 2006
Pixelwise Contrastive Loss

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

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

You want this to be large for non-matches
Pixelwise Contrastive Loss

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

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**

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

- **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

without cross-object loss

with cross-object loss

baymax
caterpillar
star_bot

baymax
caterpillar
star_bot

Silvio Savarese & Jeannette Bohg  Lecture 11
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

In this demonstration, the user clicks the right ear of the caterpillar in only one reference image.
Instance Specific Manipulation

In this demo, the user clicks on the heel of the red shoe in one reference image.

Our robot then grasps the best match for an instance-specific descriptor.
Class Consistent Manipulation

In this demo, the user clicks at the top of the laces ("the tongue") for one image of one shoe.
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 from Stereo
Supervised Learning

Finding Correspondences across Frames
Self-Supervised Learning

Monocular Depth Estimation
Unsupervised 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, ...)

\[
X \xrightarrow{\text{Data}} \xrightarrow{\text{Model}} \hat{X} \xrightarrow{\text{Data}}
\]
Learning Goals for Upcoming Lectures

Representations & Representation Learning

Optical & Scene Flow

Using Representation Learning for Depth Estimation and Finding Correspondences

Optimal Estimation

Neural Radiance Fields
CS231
Introduction to Computer Vision

Jeannette has no office hours next week.
Next two lectures (Flipped classroom):
Optical Flow and Scene Flow
Optimal Estimation