PSET 1 Part 1&2 + Project Outline

Tianyuan Dai

CS231A

04/12/2024

Anouncement

- Office Hour (Jeannette Bohg): 9am - 10am Wednesdays, Gates 244 or on zoom
- Office Hour (Krishnan Srinivasan): 3:30pm - 4:30pm Mondays, Gates 104
- Office Hour (Congyue Deng): 3:30pm - 4:30pm Tuesdays, Gates 104
- Office Hour (Tianyuan Dai): 2pm - 3pm Thursdays, Gates 315
- Room 104:

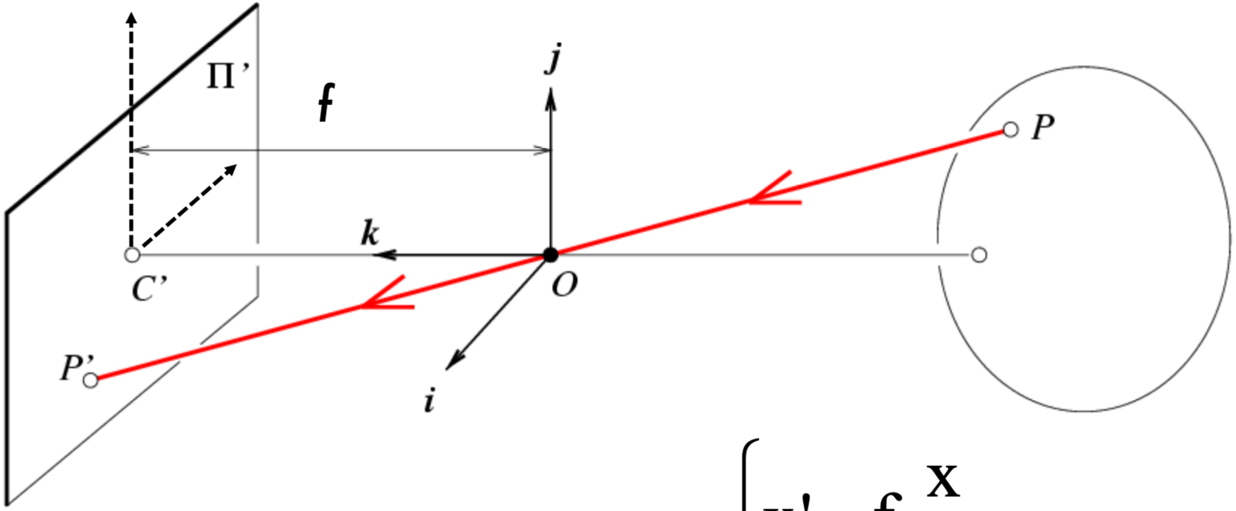
Anouncement

- For students who are looking for a project partner: Next week during all office hours (Monday, Tuesday, Thursday), for the last 15 minutes after the end of office hour, we will have project group matching for those who are looking for a partner

Overview

- Lecture Review
- PSET 1 Part 1
- PSET 1 Part 2
- Project Logistics
- Types of Projects
- Class Coverage and Ideas
- Where to Get Projects
- Helpful Resources

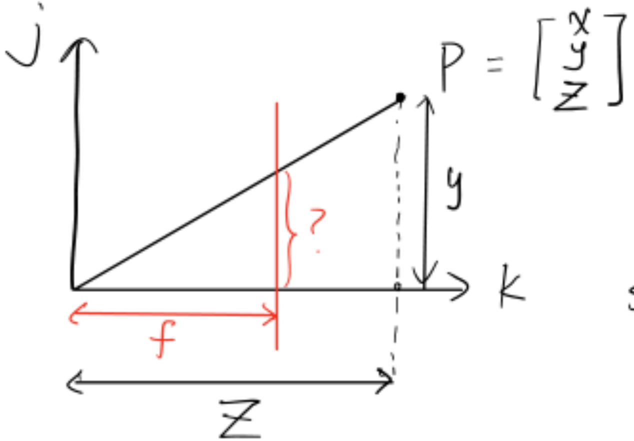
Pinhole camera



$$P = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow P' = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

$$\begin{cases} x' = f \dfrac{x}{z} \\ y' = f \dfrac{y}{z} \end{cases} \quad \text{[Eq. 1]}$$

Derived using similar triangles (Next Slide)



Similar Triangles:

$$\frac{?}{y} = \frac{f}{Z} \Rightarrow ? = f\frac{y}{z} = y'$$

similarly, $x' = f\frac{x}{z}$

Homogeneous coordinates

E→H

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

homogeneous image coordinates

$$(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

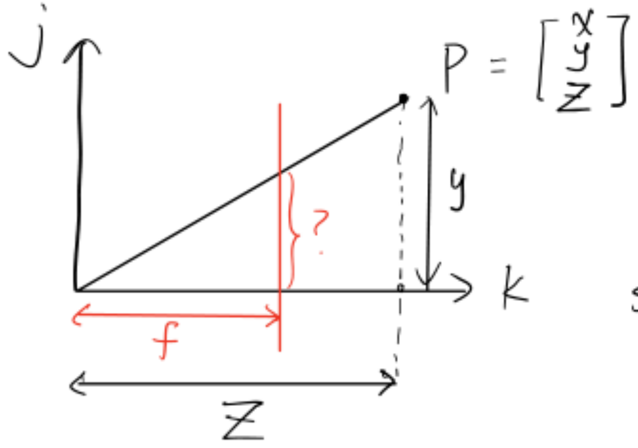
homogeneous scene coordinates

- Converting back *from* homogeneous coordinates

H→E

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$



Similar Triangles:

$$\frac{?}{y} = \frac{f}{Z} \Rightarrow ? = f\frac{y}{z} = y'$$

similarly, $x' = f\frac{x}{z}$

$$[x \quad y \quad z] \longrightarrow \left[f\frac{x}{z}, \; f\frac{y}{z}\right]$$

$$\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} fx \\ fy \\ z \end{bmatrix}$$

What are we assuming here?

1. Same coordinate systems for camera, image, and world
2. Square pixels

Projective transformation in the homogenous coordinate system

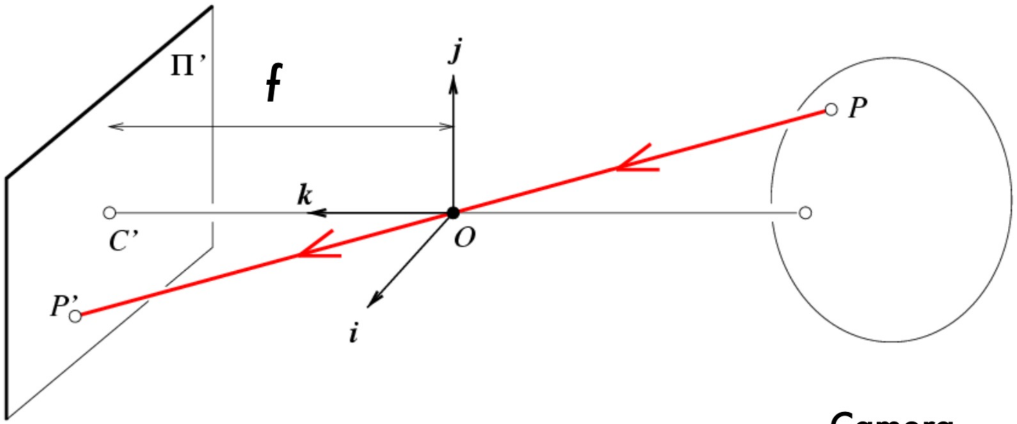
$$P_h' = \begin{bmatrix} \alpha x + c_x z \\ \beta y + c_y z \\ z \end{bmatrix} = \begin{bmatrix} \alpha & 0 & c_x & 0 \\ 0 & \beta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \underbrace{\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}}_{P_h}$$

[Eq.8]

$$M = \begin{bmatrix} \alpha & 0 & c_x & 0 \\ 0 & \beta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\overbrace{P_h'}^{\text{Homogenous}} \rightarrow \overbrace{P' = (\alpha \frac{x}{z} + c_x, \beta \frac{y}{z} + c_y)}^{\text{Euclidian}}$$

The Camera Matrix



Camera matrix K

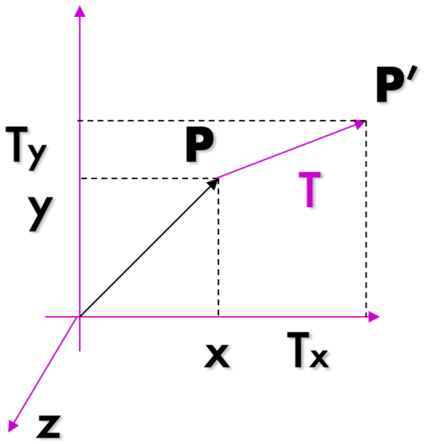
[Eq.9]

$$P' = M\,P$$

$$= K\begin{bmatrix} I & 0 \end{bmatrix} P$$

$$P' = \begin{bmatrix} \alpha & 0 & c_x & 0 \\ 0 & \beta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

3D Translation of Points



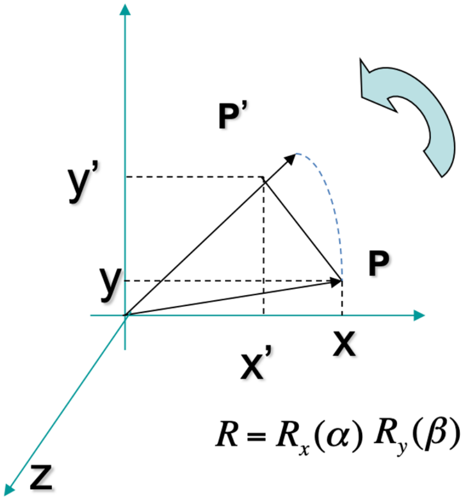
$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$$

$$P' \rightarrow \begin{bmatrix} \mathbf{I} & T \\ 0 & 1 \end{bmatrix}_{4\times 4} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

A translation vector in 3D has 3 degrees of freedom

3D Rotation of Points

Rotation around the coordinate axes, counter-clockwise:



$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

$$R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R = R_x(\alpha)\ R_y(\beta)\ R_z(\gamma)$$

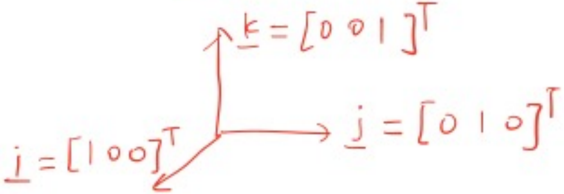
$$P' \rightarrow \begin{bmatrix} R & 0 \\ 0 & 1 \end{bmatrix}_{4\times4} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

A rotation matrix in 3D has 3 degrees of freedom

3D Rotation of Points

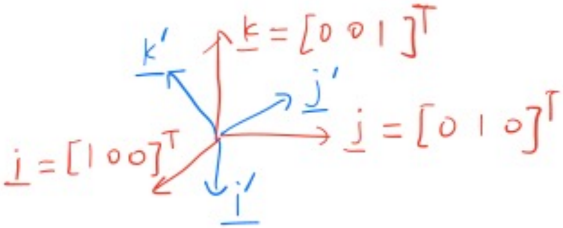
Without rotation:

$$\begin{array}{c} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \begin{array}{ccc} \underline{i} & \underline{j} & \underline{k} \end{array} \end{array} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$



Rotate:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} | & | & | \\ \text{Col 1} & \text{Col 2} & \text{Col 3} \\ | & | & | \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$



Col 1: Transformed $\underline{i}$

Col 2: Transformed $\underline{j}$

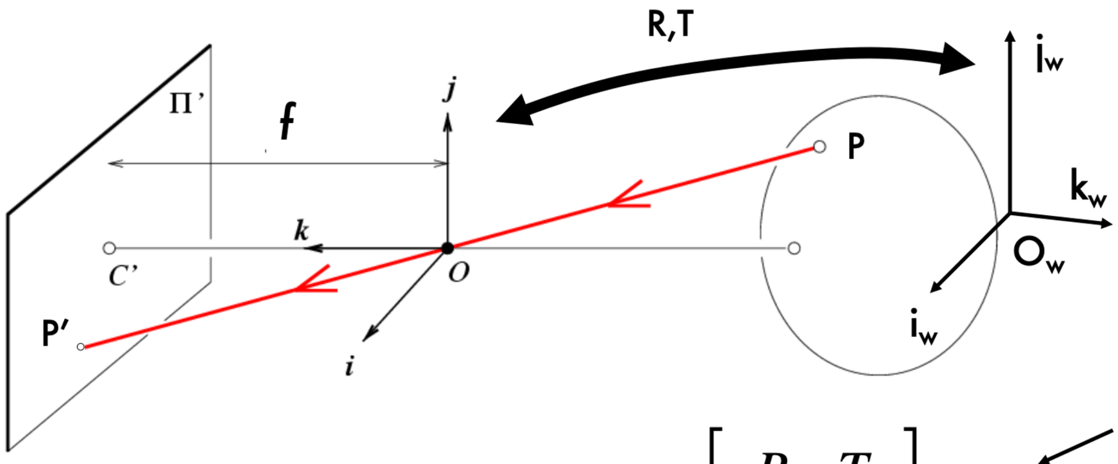
Col 3: Transformed $\underline{k}$

3D Translation and Rotation

$$R = R_x(\alpha)\ R_y(\beta)\ R_z(\gamma) \qquad T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$$

$$P' \rightarrow \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}_{4\times 4} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

World reference system



In 4D homogeneous coordinates: $P = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}_{4\times4} P_w \qquad \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$

Internal parameters

External parameters

[Eq.9]

$$P' = K\begin{bmatrix} I & 0 \end{bmatrix} P = K\begin{bmatrix} I & 0 \end{bmatrix}\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}_{4\times4} P_w = \boxed{K}\begin{bmatrix} \boxed{R \quad T} \end{bmatrix} P_w$$

[Eq.11]

Q1 - Projective Geometry

1 Projective Geometry Problems [20 points]

In this question, we will examine properties of projective transformations. We define a camera coordinate system, which is only rotated and translated from a world coordinate system.

(a) Prove that parallel lines in the world reference system are still parallel in the camera reference system. [**4 points**]

(b) Consider a unit square $pqrs$ in the world reference system where p, q, r, and s are points. Will the same square in the camera reference system always have unit area? Prove or provide a counterexample. [**4 points**]

(c) Now let's consider affine transformations, which are any transformations that preserve parallelism. Affine transformations include not only rotations and translations, but also scaling and shearing. Given some vector p, an affine transformation is defined as

$$A(p) = Mp + b$$

where M is an invertible matrix. Prove that under any affine transformation, the ratio of parallel line segments is invariant, but the ratio of non-parallel line segments is not invariant. [**6 points**]

(d) You have explored whether these three properties hold for affine transformations. Do these properties hold under any projective transformation? Justify briefly in one or two sentences (no proof needed). [**6 points**]

(a) **Prove that parallel lines in the world reference system are still parallel in the camera reference system** (note: this is an example of a rigid transformation). **[4 points]**

- Lines *k* and *l* are parallel
  - k_1 and k_2 are any two points on *k*
  - l_1 and l_2 are any two points on *l*
  - by definition of parallel lines:

$$\boxed{(k_1 - k_2) \times (l_1 - l_2) = 0}$$

$$(k_1 + p - k_2 - p) \times (l_1 + p - l_2 - p) \qquad (Rk_1 - Rk_2) \times (Rl_1 - Rl_2)$$

(b) Consider a unit square $pqrs$ in the world reference system where p, q, r, and s are points. Will the same square in the camera reference system always have unit area? Prove or provide a counterexample. [**4 points**]

- Given a square *pqrs*,

Area = $\boxed{||(q-p) \times (s-p)|| = ||q-p|| * ||s-p|| \sin\theta = 1}$

- Hint: projecting to camera frame is isometric (they preserve lengths, as well as angles between lines)
- How about affine transformation?

(c) Now let's consider affine transformations, which are any transformations that preserve parallelism. Affine transformations include not only rotations and translations, but also scaling and shearing. Given some vector p, an affine transformation is defined as

$$A(p) = Mp + b$$

where M is an invertible matrix. Prove that under any affine transformation, the ratio of parallel line segments is invariant, but the ratio of non-parallel line segments is not invariant. **[6 points]**

Consider any two parallel lines k and l. Take the segment between any two points k_1, k_2 on k and the segment between any two points l_1, l_2 on l. By definition of parallel segments,

$$k_1 - k_2 = k(l_1 - l_2)$$

for some real number k. Thus,

$$\|k_1 - k_2\| = k\|l_1 - l_2\|$$

(d) You have explored whether these three properties hold for affine transformations. Do these properties hold under any projective transformation? Justify briefly in one or two sentences (no proof needed). **[6 points]**

- Hint: Think about the difference between Projective Transformation and Affine Transformation. Briefly explain in 1 or 2 sentences.

Q2 - Affine Camera Calibration

2 Affine Camera Calibration (35 points)

In this question, we will perform affine camera calibration using two different images of a calibration grid. First, you will find correspondences between the corners of the calibration grids and the 3D scene coordinates. Next, you will solve for the camera parameters.

It was shown in class that a perspective camera can be modeled using a 3 × 4 matrix:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{1}$$

which means that the image at point (X, Y, Z) in the scene has pixel coordinates $(x/w, y/w)$. The 3×4 matrix can be factorized into intrinsic and extrinsic parameters.

An *affine* camera is a special case of this model in which rays joining a point in the scene to its projection on the image plane are parallel. Examples of affine cameras include orthographic projection and weakly perspective projection. An affine camera can be modeled as:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2}$$

which gives the relation between a scene point (X, Y, Z) and its image (x, y). The difference is that the bottom row of the matrix is $[\,0 \;\; 0 \;\; 0 \;\; 1\,]$, so there are fewer parameters we need to calibrate. More importantly, there is no division required (the homogeneous coordinate is 1) which means this is a *linear model*. This makes the affine model much simpler to work with mathematically - at the cost of losing some accuracy. The affine model is used as an approximation of the perspective model when the loss of accuracy can be tolerated, or to reduce the number of parameters being modeled. Calibration of an affine camera involves estimating the 8 unknown entries of the matrix in Eq. 2 (This matrix can also be factorized into intrinsics and extrinsics, but that is outside the scope of this homework). Factorization is accomplished by having the camera observe a calibration pattern with easy-to-detect corners.

(a) Given correspondences for the calibrating grid, solve for the camera parameters using Eq. 2. Note that each measurement $(x_i, y_i) \leftrightarrow (X_i, Y_i, Z_i)$ yields two linear equations for the 8 unknown camera parameters. Given N corner measurements, we have $2N$ equations and 8 unknowns. Using the given corner correspondences as inputs, complete the method `compute_camera_matrix()`. You will construct a linear system of equations and solve for the camera parameters to minimize the least-squares error. After doing so, you will return the 3×4 affine camera matrix composed of these computed camera parameters. In your written report, submit your code as well as the camera matrix that you compute. **[15 points]**

- Hint: `np.linalg.lstsq` can solve the least-squares solution to a linear matrix equation.

numpy.linalg.lstsq

`linalg.lstsq(a, b, rcond='warn')` [source]

Return the least-squares solution to a linear matrix equation.

Computes the vector *x* that approximately solves the equation `a @ x = b`. The equation may be under-, well-, or over-determined (i.e., the number of linearly independent rows of *a* can be less than, equal to, or greater than its number of linearly independent columns). If *a* is square and of full rank, then *x* (but for round-off error) is the “exact” solution of the equation. Else, *x* minimizes the Euclidean 2-norm $||b - ax||$. If there are multiple minimizing solutions, the one with the smallest 2-norm $||x||$ is returned.

(b) After finding the calibrated camera matrix, you will compute the RMS error between the given N image corner coordinates and N corresponding calculated corner locations in `rms_error()`. Recall that

$$\text{RMS}_{\text{total}} = \sqrt{\sum((x - x')^2 + (y - y')^2)/N}$$

Please submit your code and the RMS error for the camera matrix that you found in part (a). **[15 points]**

- Define 3D points by `real_XY`
- Project 3D points to 2D points
- Compute rms error (root mean square error)

(c) Could you calibrate the matrix with only one checkerboard image? Explain briefly in one or two sentences. **[5 points]**

- Hint: Consider when the linear system has a unique solution (consider rank of the linear system).
- What will happen if we only have one checkerboard image?

Project Logistics

- Overview here
- Teams of **1-3**: Number of people is taken into account when grading project
  - More members - More work
- Suggestions for project direction
  - Replicate an interesting paper
  - Compare different methods to a benchmark
  - Use a new approach to an existing problem
  - Implement an interesting system
  - Original research

Sharing a Project with Another Class

- Sharing projects is generally allowed
- Specify in reports
- Must be approved by both our staff and the other course staff
- Project must be profound enough that you can clarify which parts of the project were done for which class
  - Each part must be substantial enough to hold as a single project
  - Technical parts and experiments should be sufficient and different
  - For example, if you want to use CNN for flower classification, you can include some other components related to this course (e.g. geometry, …)
- Will need a separate write-up for each class

Project Grading - Important Dates

- Course project:
  - Project proposal (due Apr 25)
  - Project milestone (due May 17)
  - Poster session (June 10)
  - Final report (due June 12)

Project Proposal

- Maximum of 2 pages
- Submit the report as a PDF document through Gradescope
- Include the following:
  - Title and authors
  - Sec. Introduction: Problem you want to solve and why
  - Sec. Technical Approach: How do you propose to solve it?
  - Sec. Milestones (dates and sub-goals)
  - References
- You will be assigned a project mentor

Project Milestone Report

- Maximum of 4 pages
- Submit the report as a PDF document through Gradescope
- Include the following:
  - Title and authors
  - Sec. Introduction: Problem you want to solve and why
  - Sec. Technical Approach: How do you propose to solve it?
  - Sec. Milestones achieved so far
  - Sec. Remaining Milestones (dates and sub-goals)
  - References

Project Posters

- Announce details later.

Project Final Report

- Length of 6-8 pages
- Submit the report as a PDF document through Gradescope
- Submit your code
- Include the following:
  - Title and authors
  - Abstract
  - Sec. Introduction
  - Sec. Previous work
  - Sec. Technical Approach
  - Sec. Experiments
  - Sec. Conclusions
  - References

Class Coverage

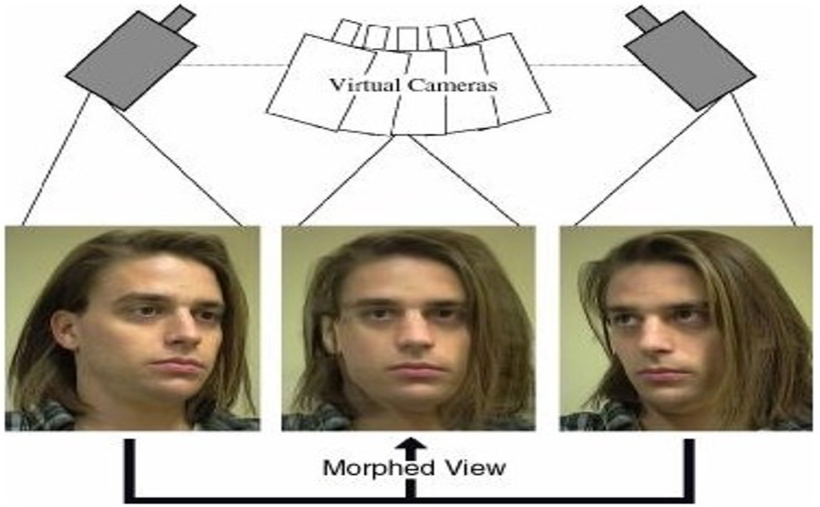
- Camera models and calibration
  - Single camera and how we model it
- Single view metrology
  - Estimating geometry from a single view
- Epipolar Geometry (Stereo Vision)
  - Estimating geometry from two viewpoints
- Structure from Motion
  - Using motion/several viewpoints to estimate structure
- Volumetric Stereo
  - Using multiple views to map 3D points

Class Coverage

- Representations and Representation Learning
  - Extracting features from 2D images for downstream applications
- Monocular Depth Estimation & Feature Tracking
  - Estimating depth in images, tracking of pixels in videos
- Optical and Scene Flow
- Optimal Estimation
- Neural Radiance Fields

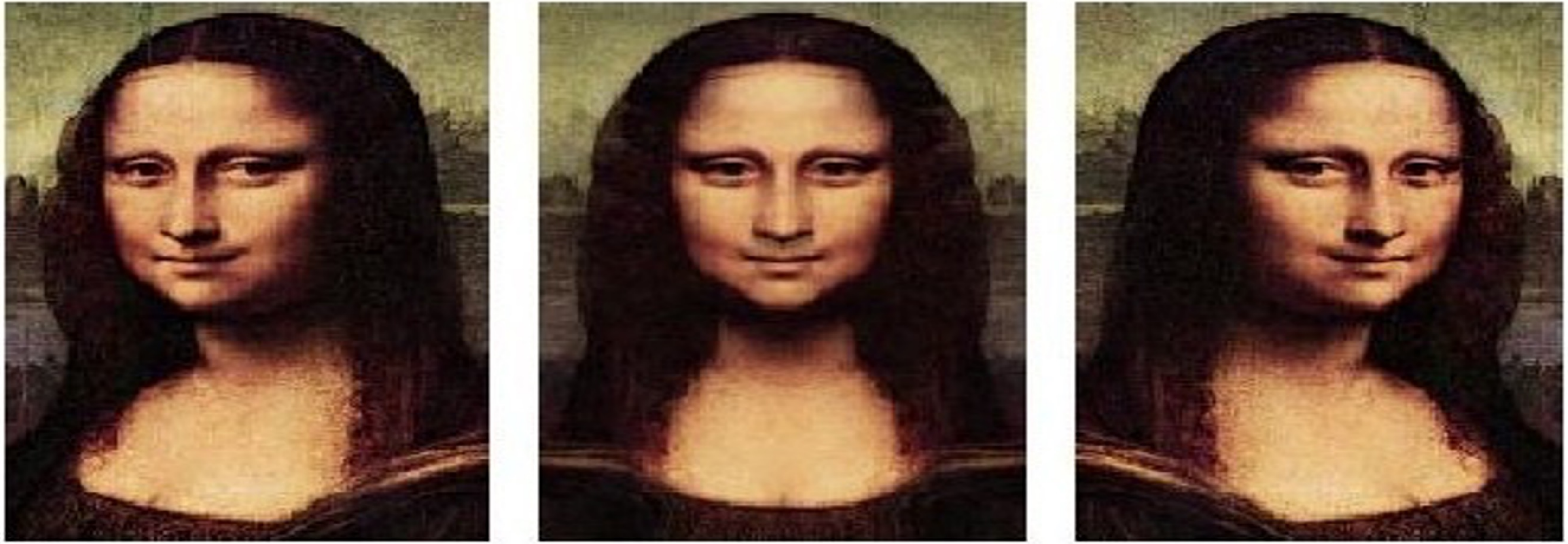
View Morphing

Image morphing techniques can generate compelling 2D transitions between images.



S. M. Seitz and C. R. Dyer, *Proc. SIGGRAPH 96*, 1996, 21-30

View Morphing



Automatic Photo Pop-Up

A fully automatic method for creating a 3D model from a single photograph

Hoiem, D., Efros, A. A., and Herbert, M, “Automatic Photo Pop-Up”, SIGGRAPH 2005.

Novel Hardware

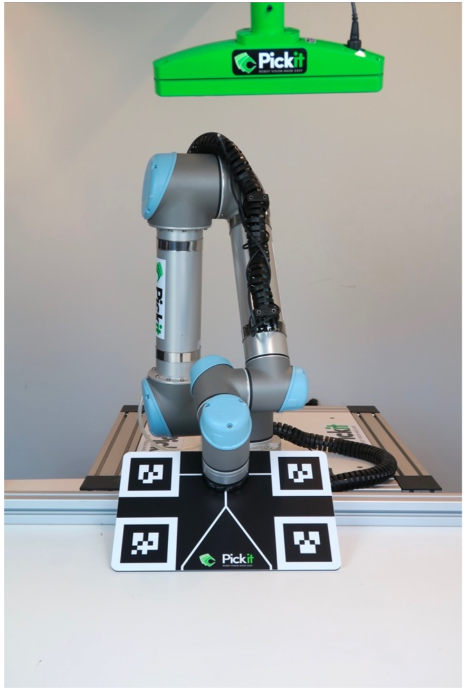
Mobile Devices

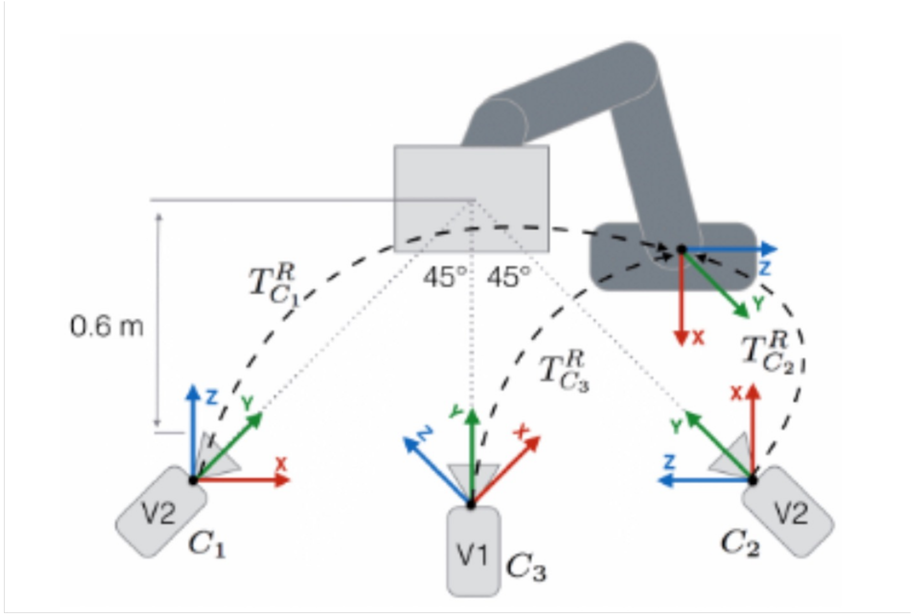
Can you take an existing vision algorithm and adapt it to a mobile device to make it more useful?

Monocular Depth estimation

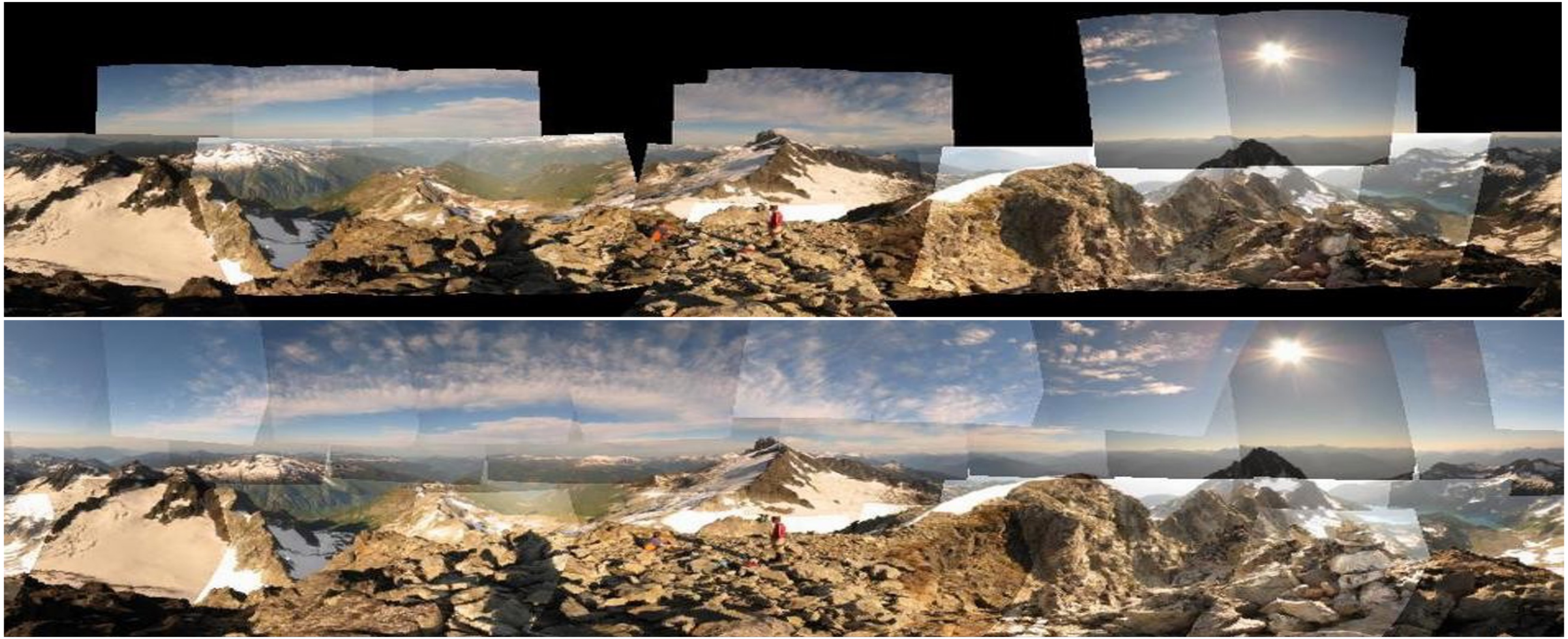
Figure 10. Snapcode of publicly available community lens

Camera Calibration for Robots





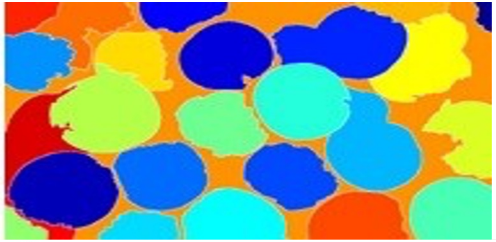
Recognizing Panoramas

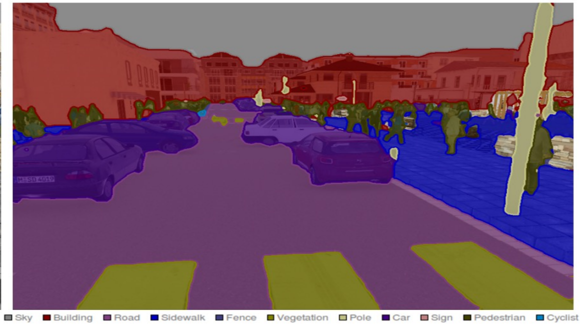


Brown, M. and Lowe, D. G., “Recognizing Panoramas”, ICCV 2003.

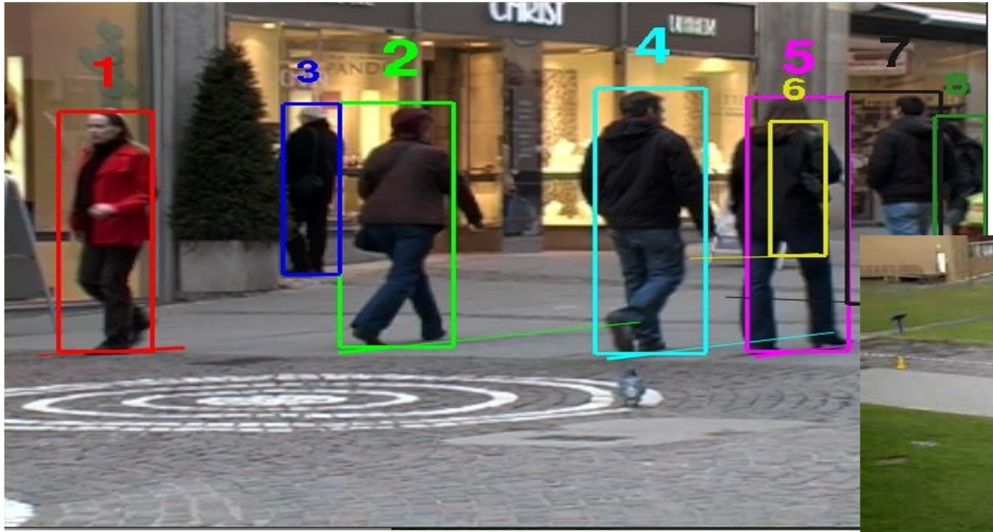
Image Segmentation

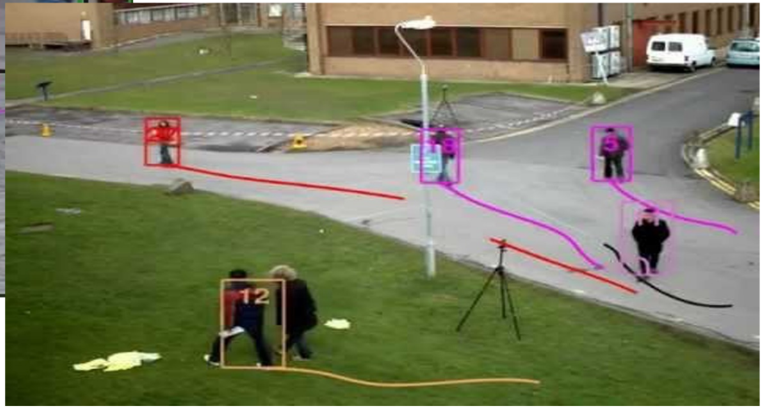
Partition an image into multiple segments (sets of pixels) in order to make it easier to analyze



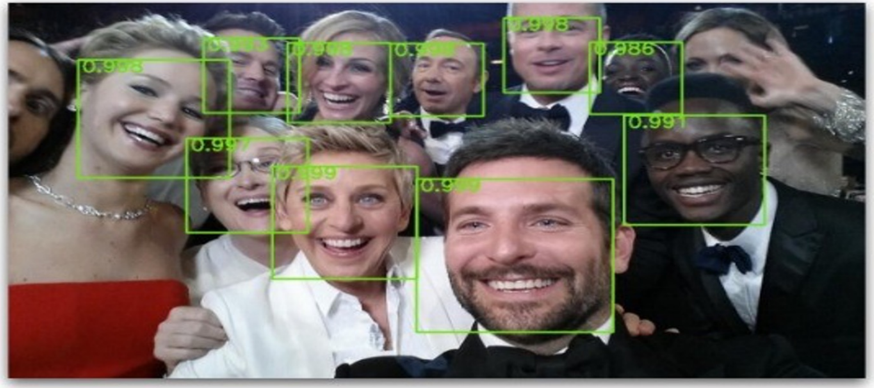


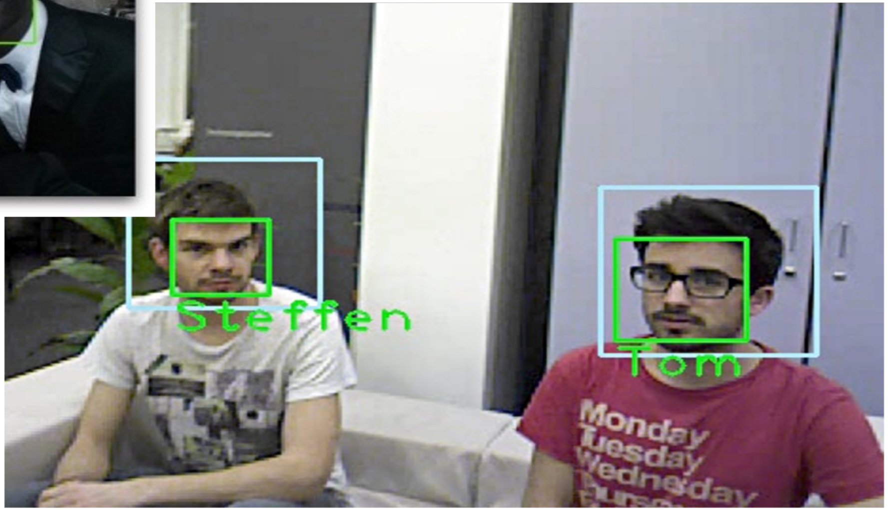
Tracking





Face Detection – Face Identification





Other Topics

- Pose Estimation: Estimate the skeleton angles for a person from an image/video
- Action and Gesture Recognition: Is a person standing, walking, or sitting in an image/video? Is he/she waving?
- Scene Understanding: Can you classify a scene? Can you recognize and/or segment each component of the scene?
- Trajectory Forecasting
- ...

Negative project examples

- Projects without components related to the course
- Applying Alexnet for image classification
- Finding and running an existing Github code
- Only running OpenCV libraries for a task
- …

Where to get Project Ideas

- Course Staff: Office hours, ideas posted on website
- Computer vision papers and conferences
  - CVPR
  - ICCV
  - ECCV
- Computer vision research groups at Stanford
  - Silvio Savarese
  - Fei-Fei Li
  - Juan Carlos Niebles
- Past projects: See course website
- Papers with Code: Computer Vision
- Come up with your own!

Datasets

- Many are available on the web
- See the following aggregators:
  - CV Datasets on the Web
  - Yet Another Computer Vision Index To Datasets (YACVID)
- References found in papers
- Course CA’s

Project Advice

- Choose your team well
- Make sure the scope of your project fits a quarter
  - Set a minimum goal, desired goal, and a moonshot
- Constrain your problem smartly
- See what datasets are available if you are doing a recognition project
  - Specially for deep learning projects
- You may need to plan ahead/learn outside materials
- Use software when available
  - OpenCV, MATLAB, Deep learning frameworks
- Come ask questions – We’re happy to talk!

Thanks!

Questions