INTRODUCTION TO COMPUTER VISION

Lecture 7 – Recap & Overview

Gyeongsik Moon

Visual Computing and AI Lab

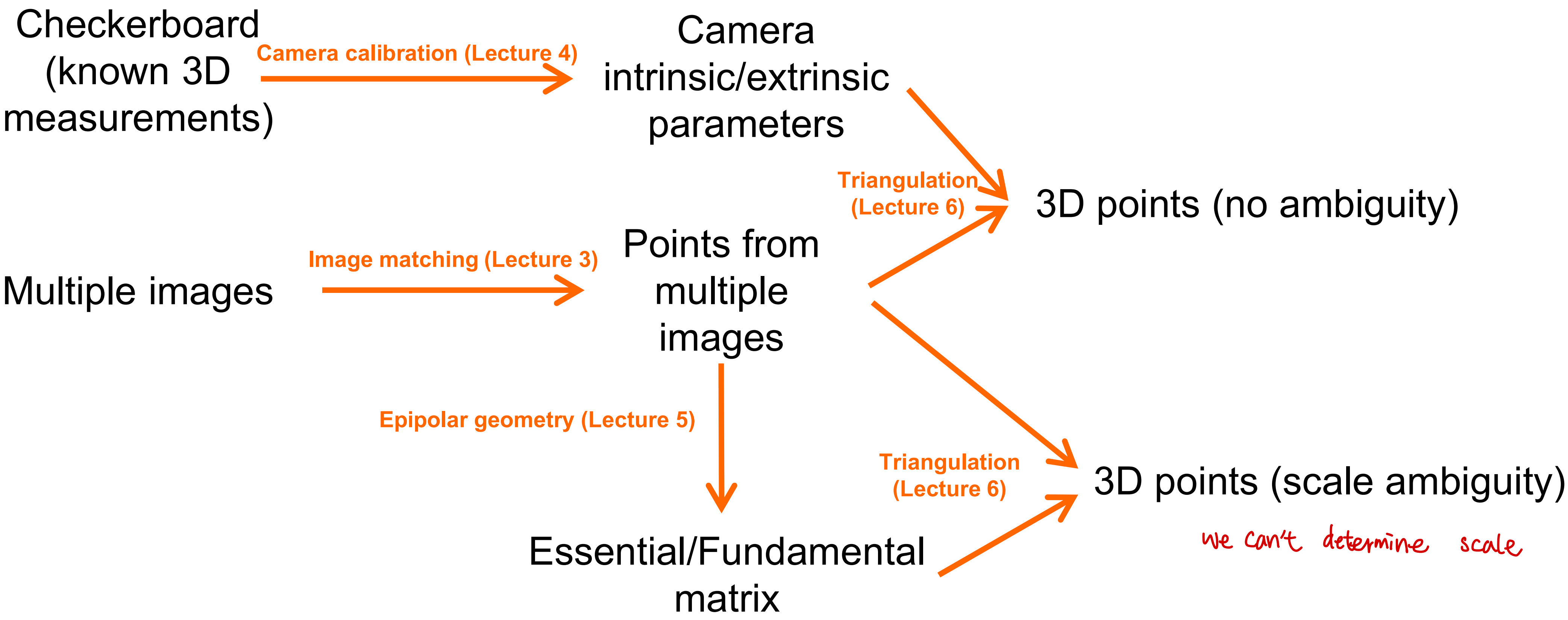
Korea University

Recap and Overview

- We’ve covered lots of mathematical stuff
- Let’s take a break and recap what we’ve learned so far
- I hope you won’t get lost, as the many slides can make it hard to find direction

- All the entire pipeline: multi-view geometry theory

Slide from Lecture 5



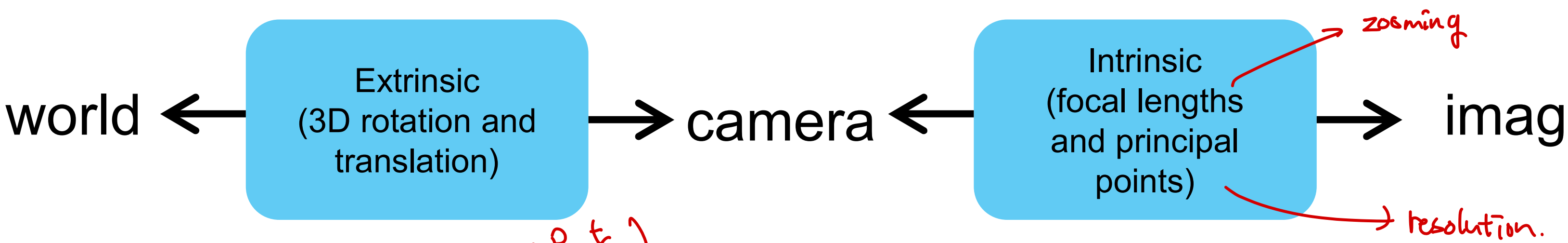
we can't determine scale

Homogeneous Coordinates (Lecture 2)

- Introduce an additional scalar to represent coordinates
- We can explain perspective distortion in the homogeneous space
- We can represent all projections only with matrix multiplication
    - Linear system → computation much easier

Coordinate systems (Lecture 2)

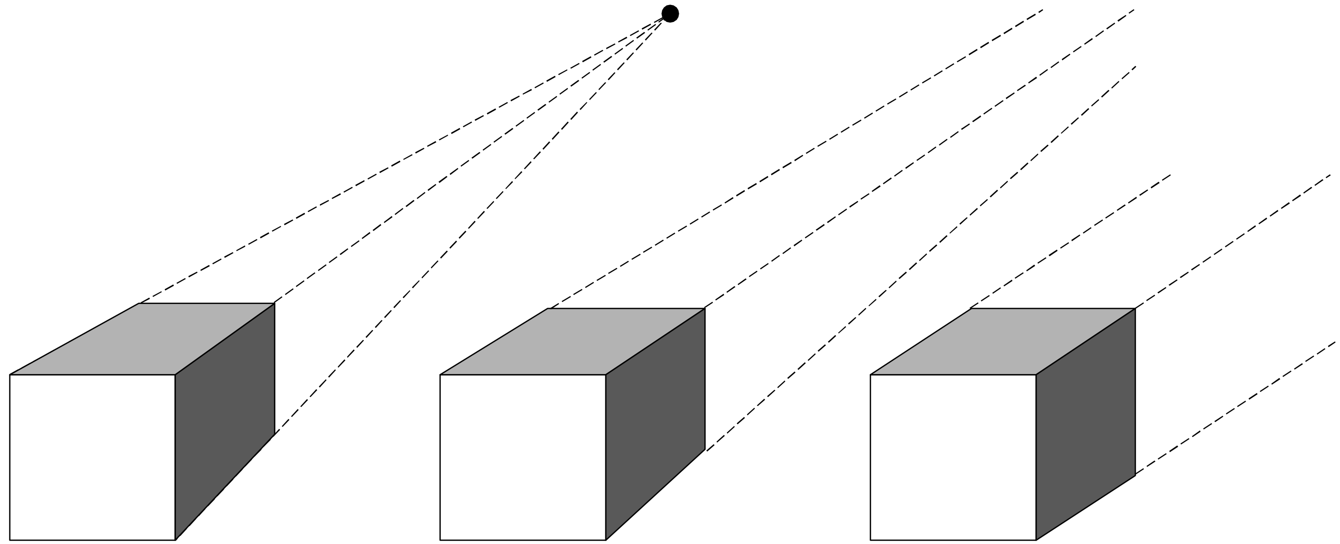
- World coordinate system (3D)
  - A reference (canonical) coordinate system
  - Fixed coordinate system
  - You can define your own one
- Camera coordinate system (3D)
  - Defined for each camera
  - Camera-relative coordinate system
- Image coordinate system (2D)
  - Defined for each camera
  - Projected space from the camera coordinate system



$\begin{pmatrix} R & t \\ 0^T & 1 \end{pmatrix}$

Projection Models

Slide from Lecture 2



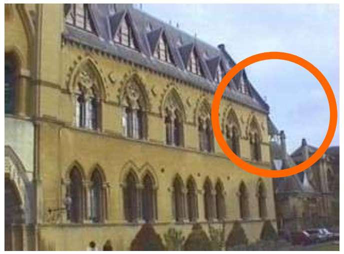
Perspective

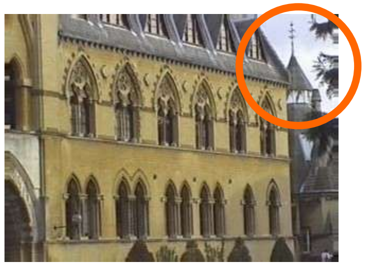
Weak Perspective /Orthographic

Realistic
Harder to get

Increasing Focal Length / Distance from Camera

Unrealistic
easier





All objects appearance.

Different from simply zooming in the 2D image space

Summary

Slide from Lecture 2

- World coordinates (Blue)
  - A reference 3D coordinate system
- Camera coordinates (Orange)
  - Defined in the 3D space for each camera
- Image coordinates (Black)
  - Defined in the 2D space for each camera

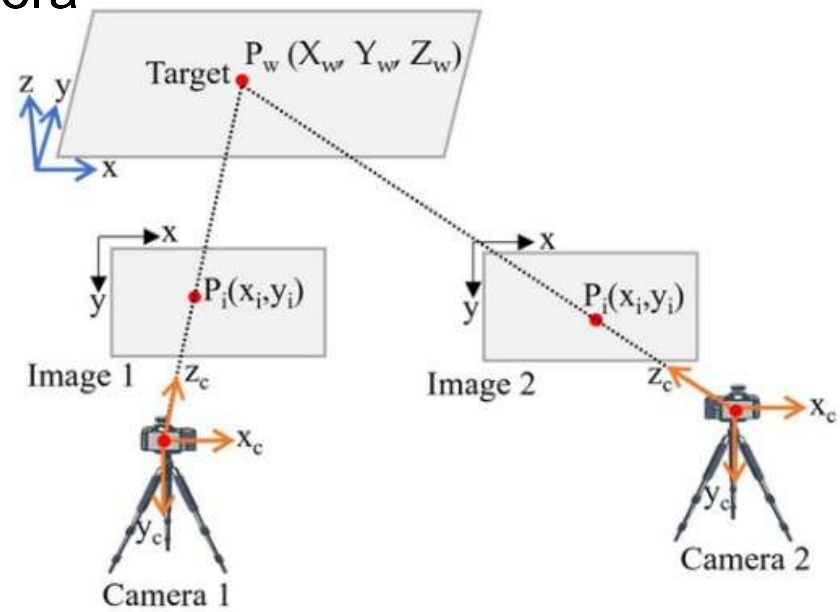


Image Matching (Lecture 3)

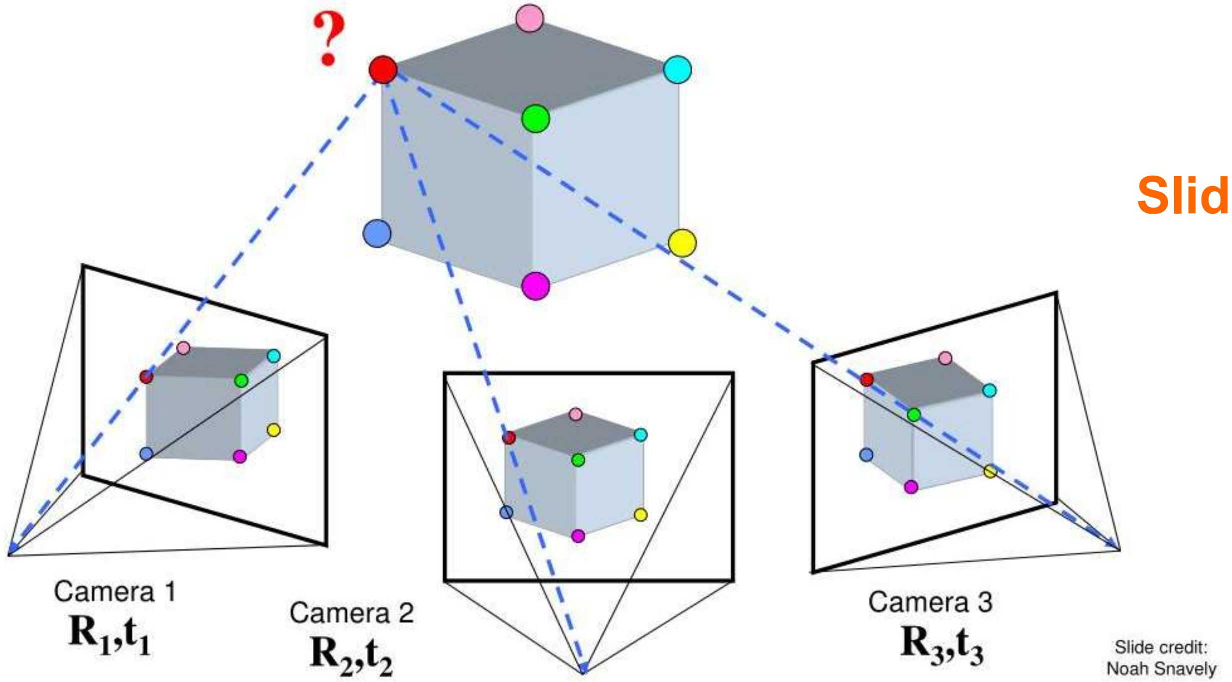
- Find correspondences between images
- Detect repeatable and distinctive features
  - SIFT
- Find the closest matching between detected features

Multi-View Geometry (MVG) Theory

According to MVG (we'll learn this later), if we know

- Camera intrinsic/extrinsic parameters (we learnt what are they in prev. classes)
- *Matched points across multiple viewpoints (same-colored dots in images)*

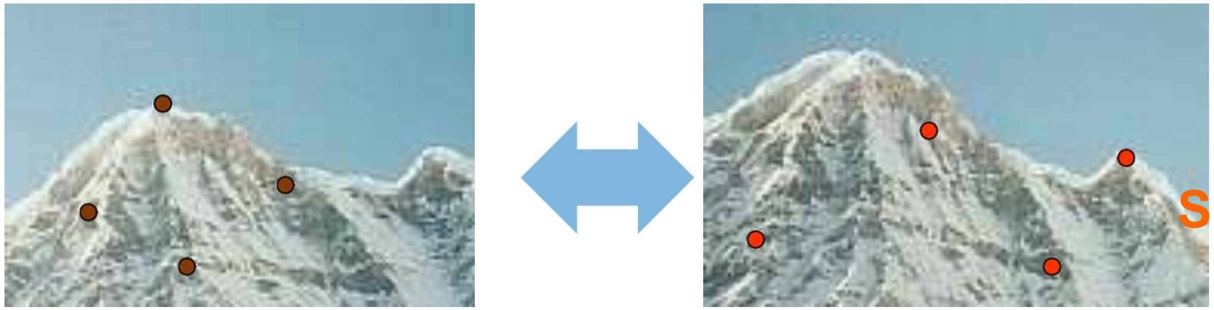
, then, we can lift the multi-view observations to the 3D space



Slide from Lecture 3

Matching with Features

Problem 1: How to **detect** the **same** points **independently** in both images?



Slide from Lecture 3

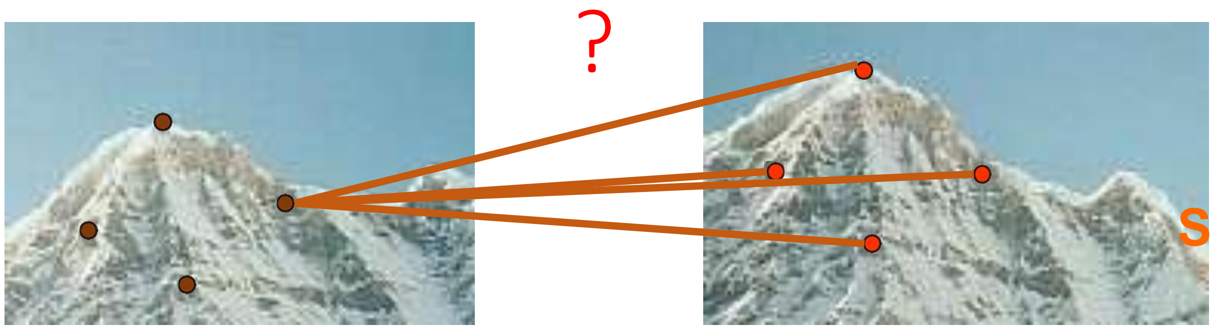
no chance to match!

We need a **repeatable** feature **detector.** Repeatable means that the detector should be able to re-detect the same feature in different images of the same scene.

This property is called **Repeatability** of a feature **detector**.

Matching with Features

Problem 2: For each point, how to **match** its **corresponding point** in the other image



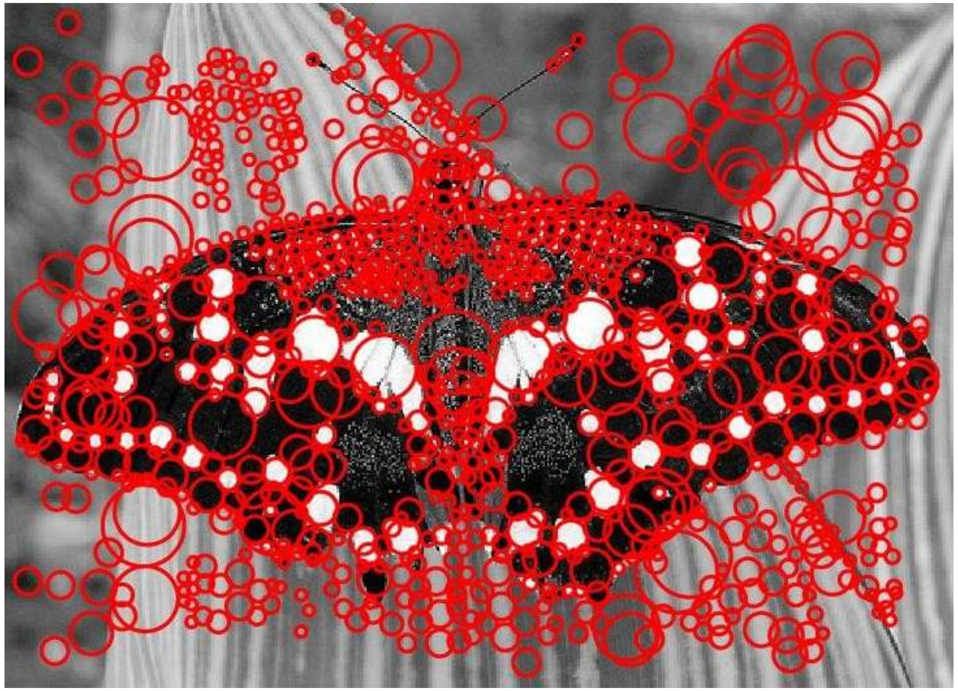
Slide from Lecture 3

We need a **distinctive** feature descriptor. A descriptor is a "description" of the pixel information around a feature (e.g., patch intensity values, gradient values, etc.). Distinctive means that the descriptor uniquely identifies a feature from other features without ambiguity. This property is called **Distinctiveness** of a feature **descriptor**.

The descriptor must also be **robust to geometric and photometric** changes.

Local extrema of DoG images across Scale and Space

- Some *distinct* points can be extracted from DoG (differences of Gaussians)
  - Local patches without distinct textures should not have big differences
- For the visualization, draw a circle at the position of the local extrema where the radius of the circle is from selected scale (dominant rotations are not included for the visualization)



Slide from Lecture 3

How it is implemented in practice

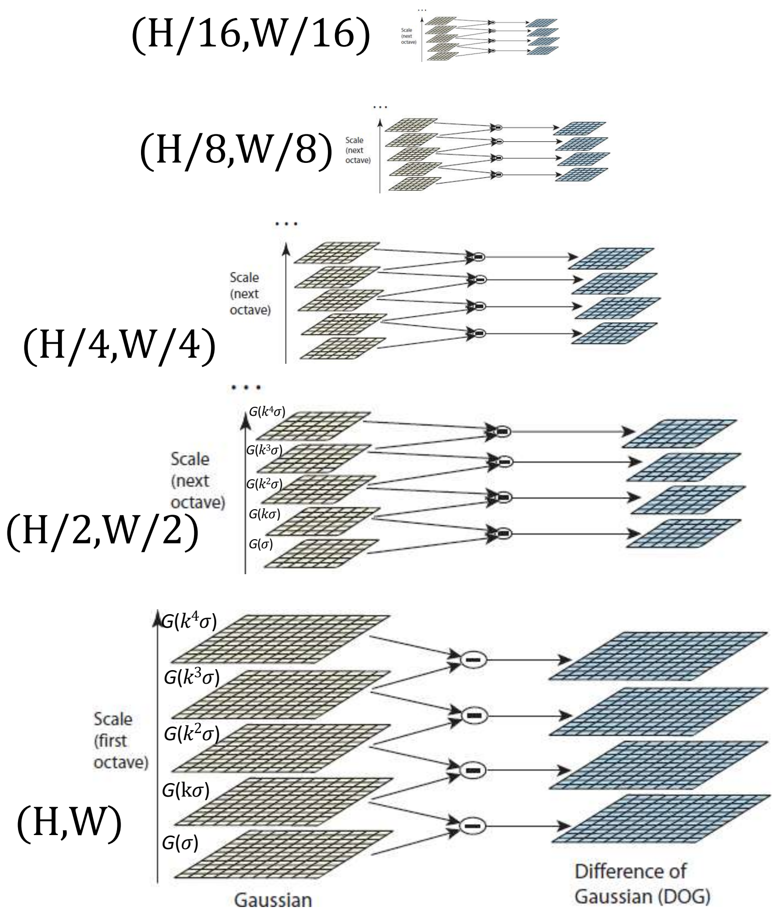
Slide from Lecture 3

1. Build a Space-Scale Pyramid:

- The initial image is incrementally convolved with Gaussians $G(k^i\sigma)$ to produce blurred images separated by a constant factor k in scale space (shown stacked in the left column).
  - The initial Gaussian G(σ) has σ=1.6
  - k is chosen: $k = 2^{1/s}$, where s is the number of intervals into which each octave of scale space is divided
  - Each octave consists of s images, blurred with different stds.
- Adjacent blurred images are then subtracted to produce the **Difference-of-Gaussian (DoG)** images

2. Scale-Space extrema detection

- Detect local maxima and minima in space-scales

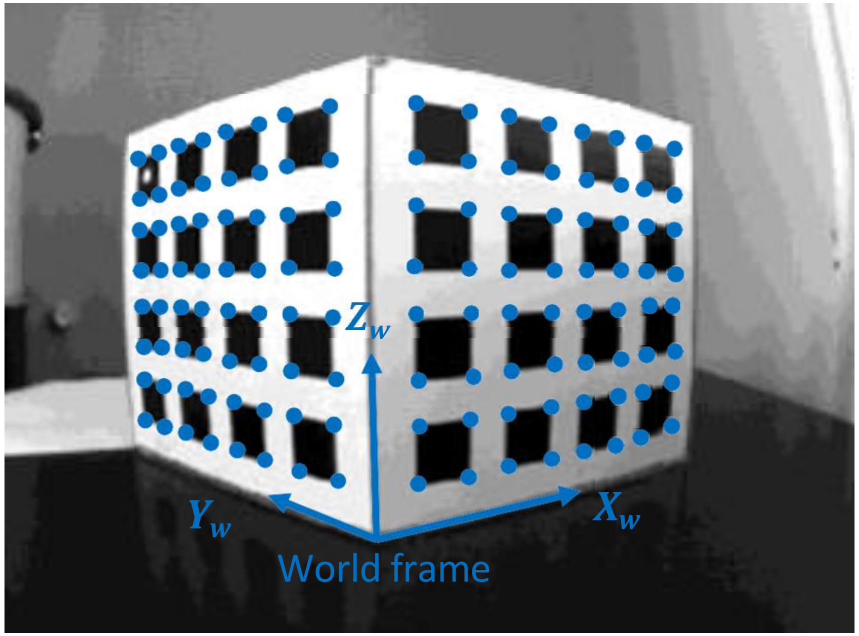


Camera Calibration (Lecture 4)

- Given 1) matched 2D points and 2) 3D measurements, find camera intrinsic/extrinsic parameters
- Often use checkerboards
  - Easy to detect 2D points
  - Know actual 3D sizes and positions

Tsai's Method: Calibration from 3D Objects

- This method was proposed in 1987 by Tsai and consists of measuring the 3D position of $\boldsymbol{n \geq 6}$ **control points** on a 3D calibration target and the **2D coordinates of their projection** in the image.
- Assumption: we know 2D and 3D coordinates of control points
  - 2D: image pre-processing (e.g., corner detectors)
  - 3D: we know actual size of the 3D object and actual positions of control points as well



Slide from Lecture 4

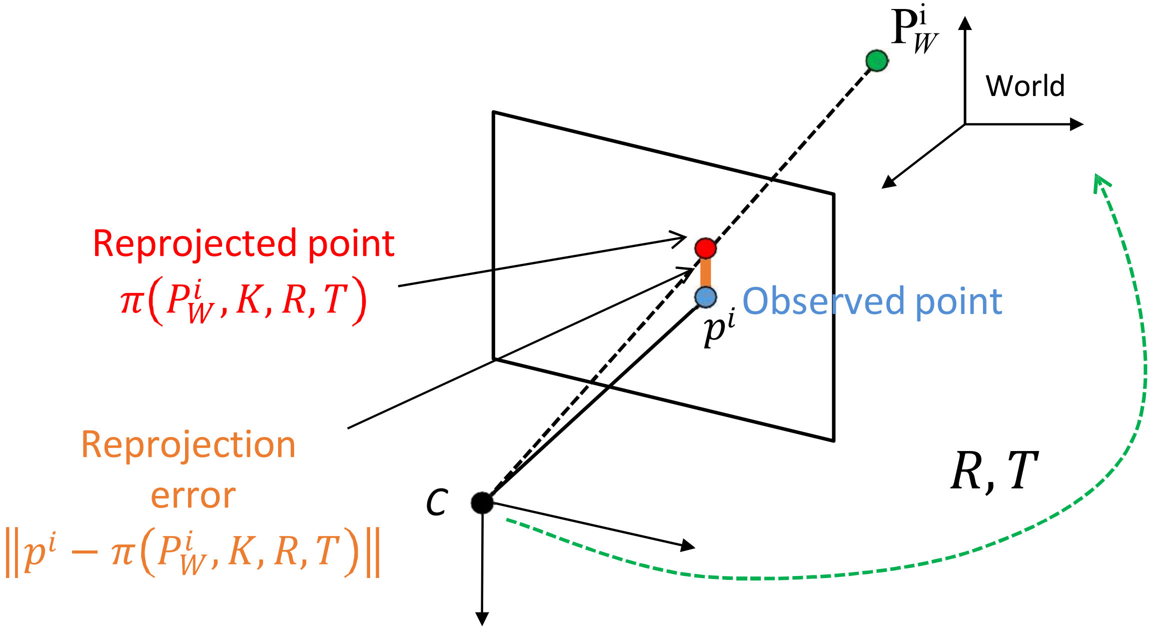
Tsai, Roger Y. (1987) "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses,'' *IEEE Journal of Robotics and Automation*, 1987. PDF.

Reprojection Error

accuracy
보통 에러도 풀리가 있어서 RANSAC 사용.
여기서의 가정은 에러가 존재.

- The reprojection error is the **Euclidean distance** (in pixels) between an **observed image point** and the **corresponding** 3D point **reprojected** onto the camera frame.
- The reprojection error gives us a **quantitative measure of the accuracy** of the calibration (**ideally it should be zero**).

Slide from Lecture 4



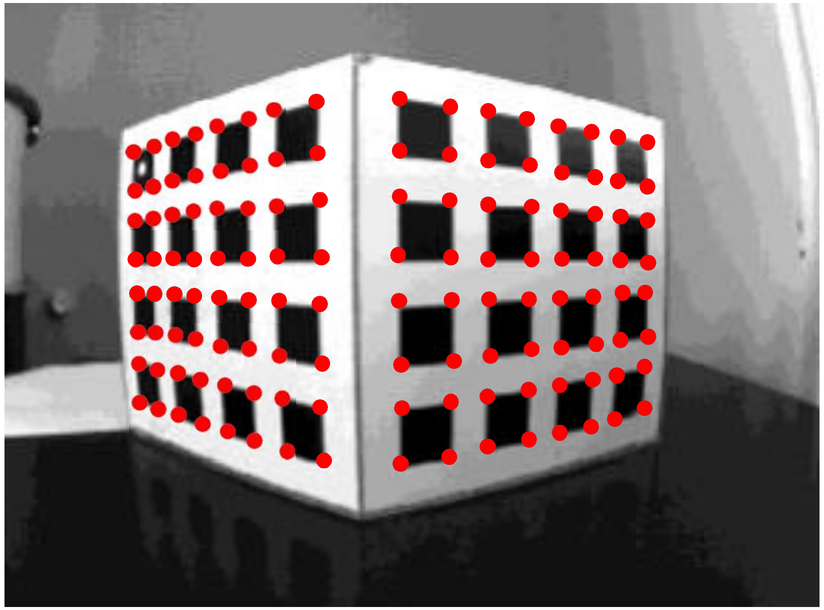
Non-Linear Calibration Refinement

Slide from Lecture 4

- The calibration parameters K, R, T determined by the DLT can be refined by minimizing the following cost:

$$K, R, T, lens\ distortion = argmin_{K,R,T,lens} \sum_{i=1}^{n} \left\| p^i - \pi\left(P_W^i, K, R, T\right) \right\|^2$$

- This time we also include the **lens distortion** (can be set to 0 for initialization)
- Can be minimized using **Levenberg–Marquardt** (more robust than Gauss-Newton to local minima)



● Control points (observed points)

● Reprojected points $\pi\left(P_W^i, K, R, T\right)$

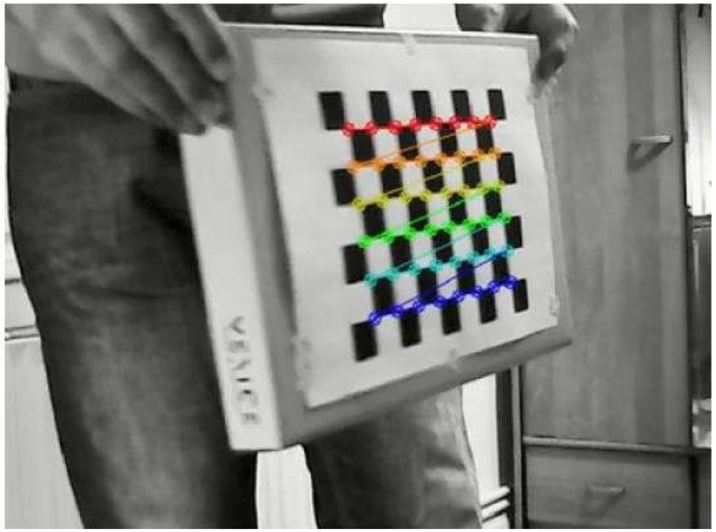
DLT → RE refine

DLT skip이 안되는 이유. K,R,T를 random initialize 하면 수렴이 안되거나 발산할 수 있음.

초기값이 중요.

Zhang's Algorithm: Calibration from Planar Grids

- **Tsai's calibration** requires that the world's 3D points are non-coplanar, which is **not very practical**
- **Today's camera calibration toolboxes** (Matlab, OpenCV) use **multiple views** of a **planar grid** (e.g., a checker
- board)
- They are based on a method developed in 2000 by Zhang (Microsoft Research)

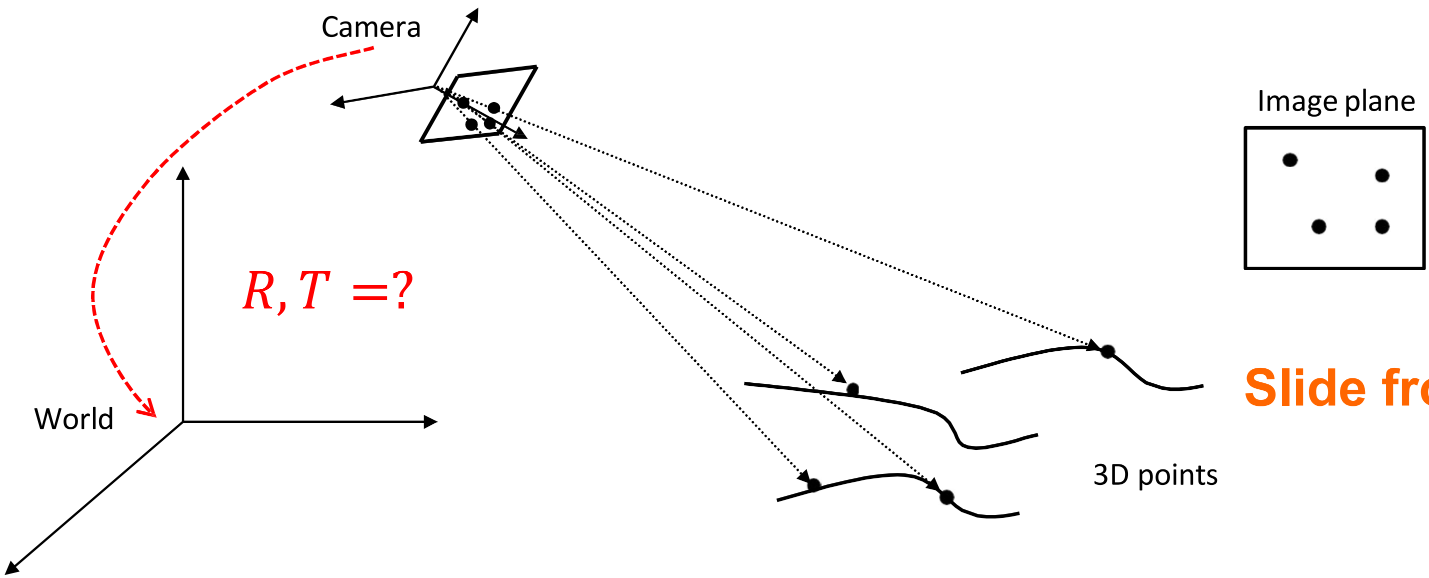


Slide from Lecture 4

Zhang, A flexible new technique for camera calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000. PDF.

Camera Localization (or Perspective from n Points: PnP)

- This is the problem of determining the **6DoF pose of a camera** (position and orientation) with respect to the world frame **from a set of 3D-2D point correspondences**.
- It assumes the **camera** to be **already calibrated**
- **In other words, the goal is getting extrinsics (R and T) while intrinsics are given**
- The **DLT can be used** to solve this problem **but is suboptimal**. We want to study **algebraic solutions** to the problem.



Slide from Lecture 4

Epipolar Geometry (Lecture 5)

2 view

- Given matched 2D points, find camera intrinsic and extrinsic parameters (with scale ambiguity)
- Often used when we do not have checkboards but still need camera parameters

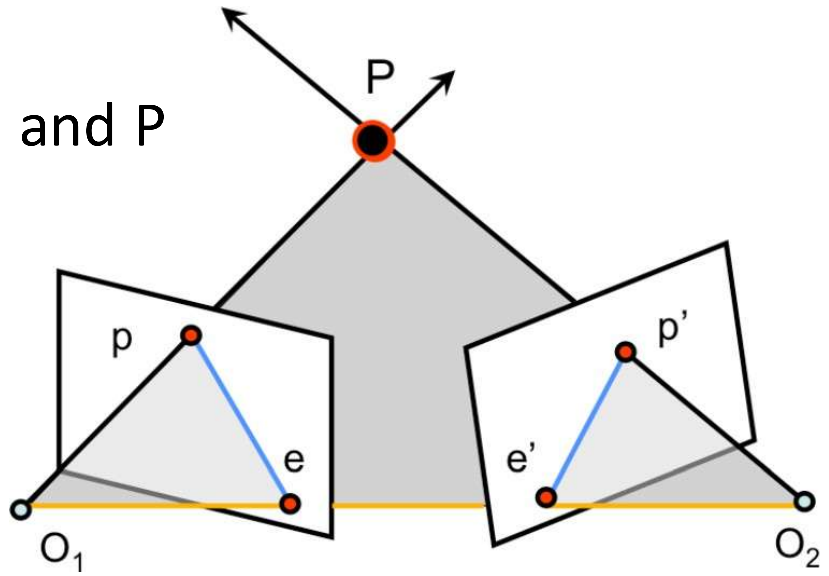
Camera Calibration vs. Epipolar Geometry

- If we know extrinsics with camera calibrations and checkerboard
  - E.g., you calibrate cameras with checkerboards
  - Skip all things we'll learn today
  - Just triangulate (we'll learn this later) 2D points to the 3D space
- If we do not know extrinsics (today's focus)
  - E.g., taking a video with your mobile phone without any calibrations/checkerboards
  - Epipolar geometry!
  - We get essential/fundamental matrices
  - We do not have checkerboard, which provides actual 3D measurements -> we get extrinsics up to scale

Slide from Lecture 5

Epipolar Geometry

- Baseline (Yellow line)
  - The line between the two camera centers O_1 and O_2
- Epipolar plane (gray plane)
  - Defined by P, O_1, and O_2; contains baseline and P
- Epipoles (e and e')
  - ∩ of baseline and image plane
  - Projection of the other camera center
- Epipolar lines (Blue lines)
  - ∩ of epipolar plane with the image plane



Slide from Lecture 5

Epipolar Constraint

- Essential matrix vs. Fundamental matrix
  - Similarity   about camera parameter
    - Both relate the matching image points
      - – Encode epipolar geometry of two views & camera parameters
  - Differences
    - E (essential matrix) encodes only the camera extrinsic parameter   minimal necessary information.
    - F (fundamental matrix) also encodes the intrinsic parameters

$$\mathbf{p}'^{T} E \mathrm{p} = 0$$
$$E = [\mathbf{t}_{\times}]R$$

Essential matrix

$$\mathbf{p}'^{T} F \mathrm{p} = 0$$
$$F = K'^{-T}[\mathbf{t}_{\times}]RK^{-1}$$

Fundamental matrix

Slide from Lecture 5

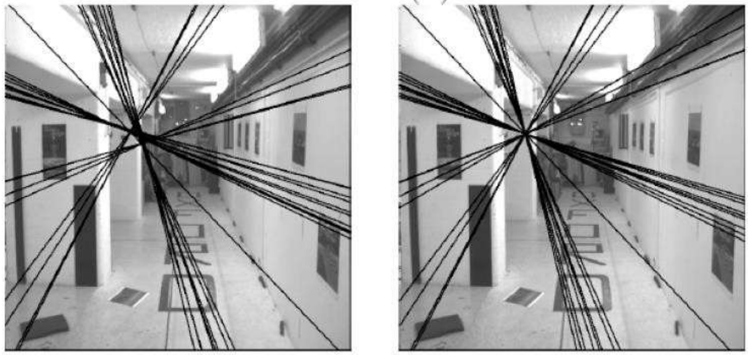
Epipolar Constraint

- Properties of the Fundamental matrix
  - 3 by 3
  - homogeneous (has scale ambiguity)
  - rank(F) = 2
    - The potential matching point is located on a line
  - F has 7 degrees of freedom (3x3 – 1 (rank2) – 1 (scale ambiguity) = 7)

$$\mathbf{p}'^T F \mathrm{p} = 0 \qquad F = K'^{-T}[\mathbf{t}_{\times}]RK^{-1}$$

Slide from Lecture 5

Fundamental matrix has rank 2 : $\det(\mathtt{F}) = 0$.



Left : Uncorrected $\mathtt{F}$ – epipolar lines are not coincident.

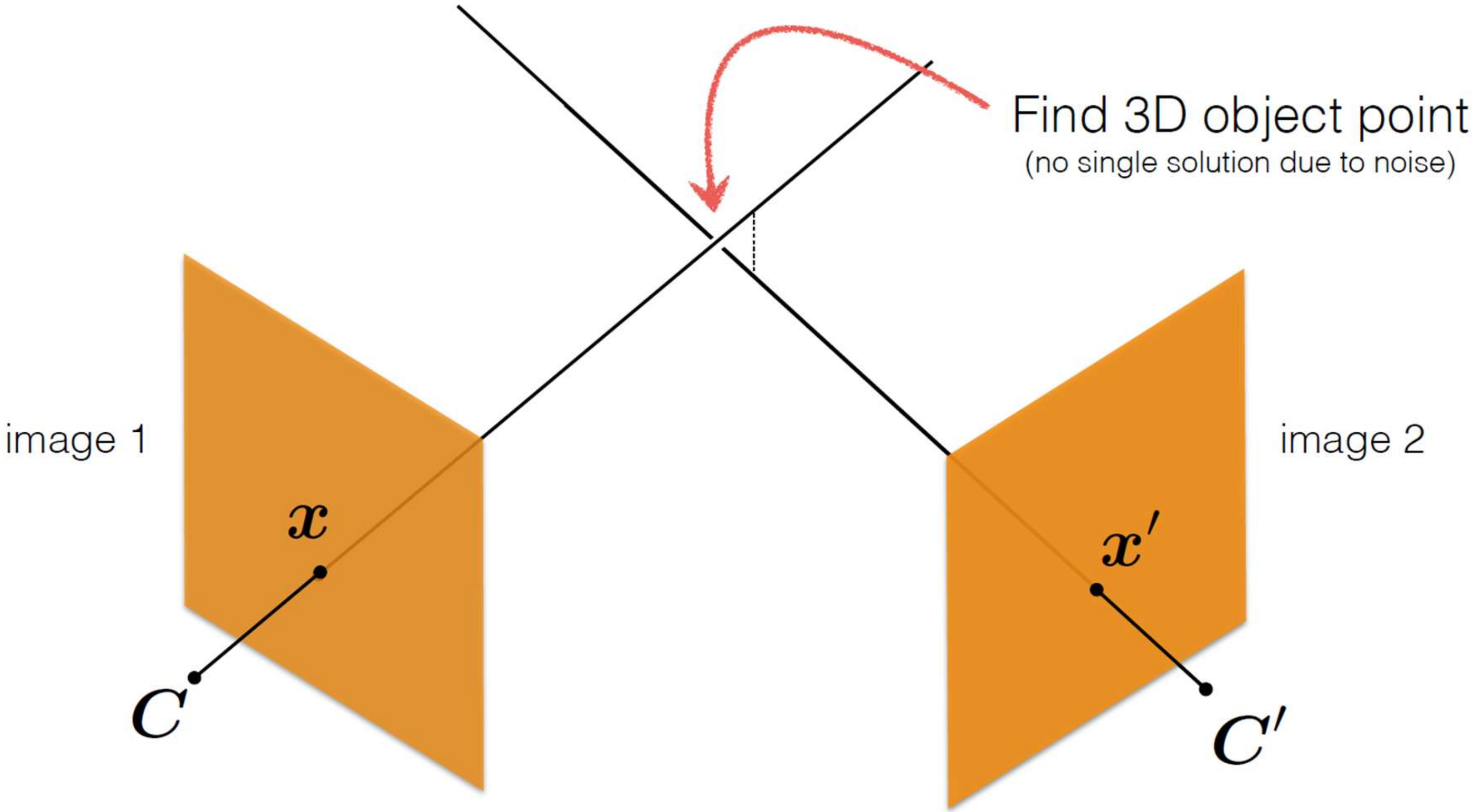
Right : Epipolar lines from corrected $\mathtt{F}$.

Triangulation and Bundle Adjustment (Lecture 6)

- Triangulation
  - Given 1) matched 2D points and 2) camera parameters, lift the 2D points to the 3D space
  - DLT is used   콤비네이션 of DLT & SVD
- Bundle adjustment
  - Further optimize camera parameters and 3D coordinates based on 2D matching results
  - RANSAC is used to reject outliers

Triangulation

Slide from Lecture 6



Bundle Adjustment

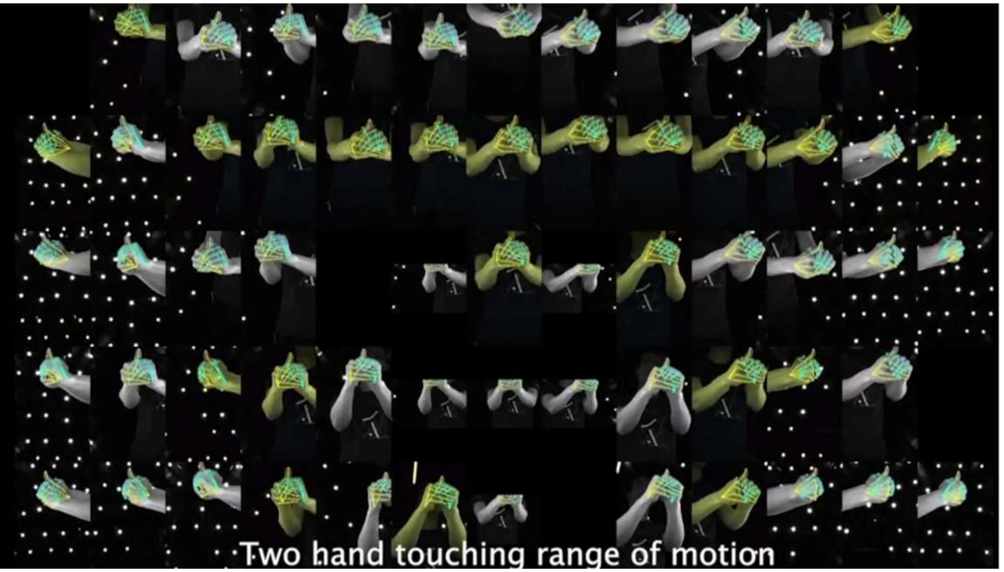
Slide from Lecture 6

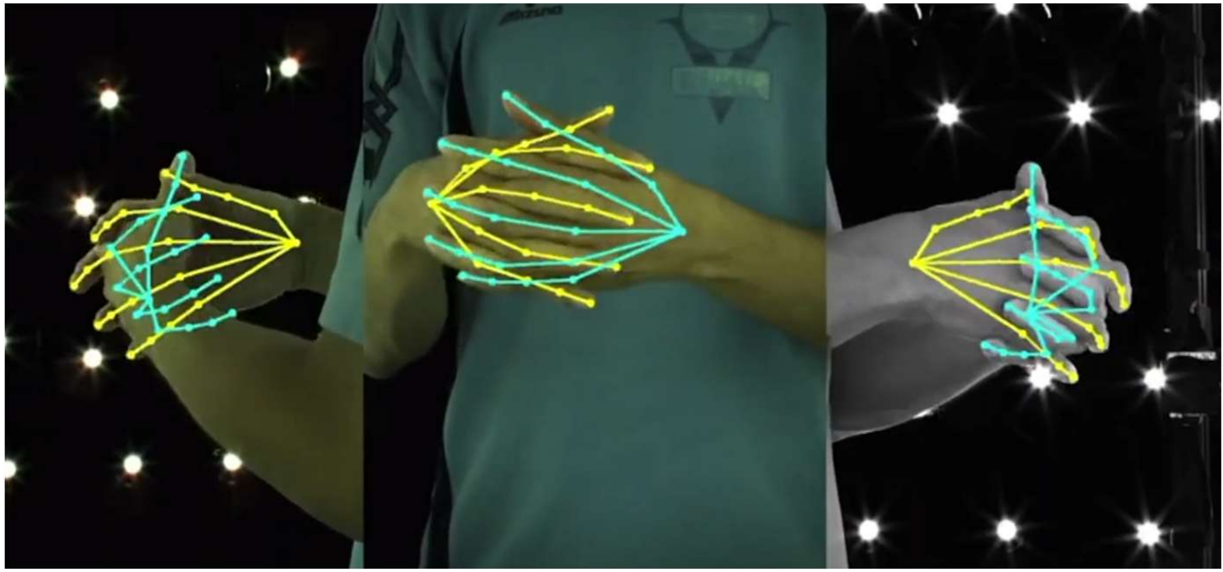
- From initial triangulated 3D points, jointly optimizing 3D coordinates and camera parameters by minimizing the reprojection error

- Similar to the non-linear calibration refinement of Lecture 4

RANSAC for 3D lifting

Slide from Lecture 6



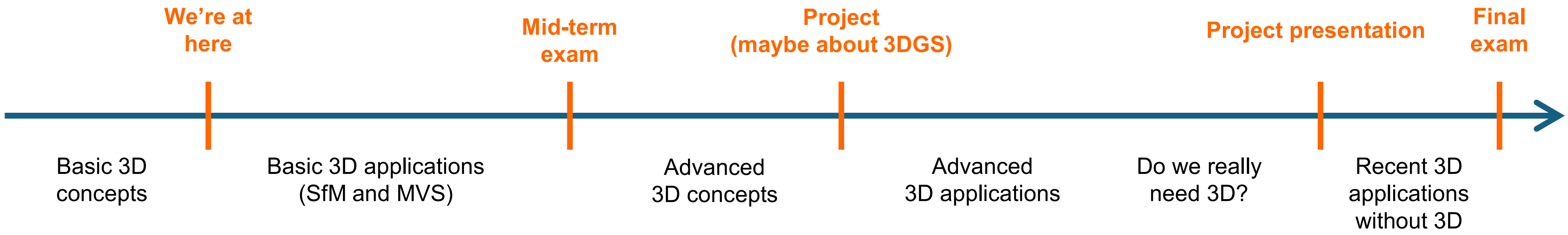


Algorithm:

1. Sample (randomly) the number of points required for the triangulation
2. Triangulate the selected 2D points to the 3D space
3. Project the triangulated 3D points to all image space and check reprojection error. Reject viewpoints with huge error.

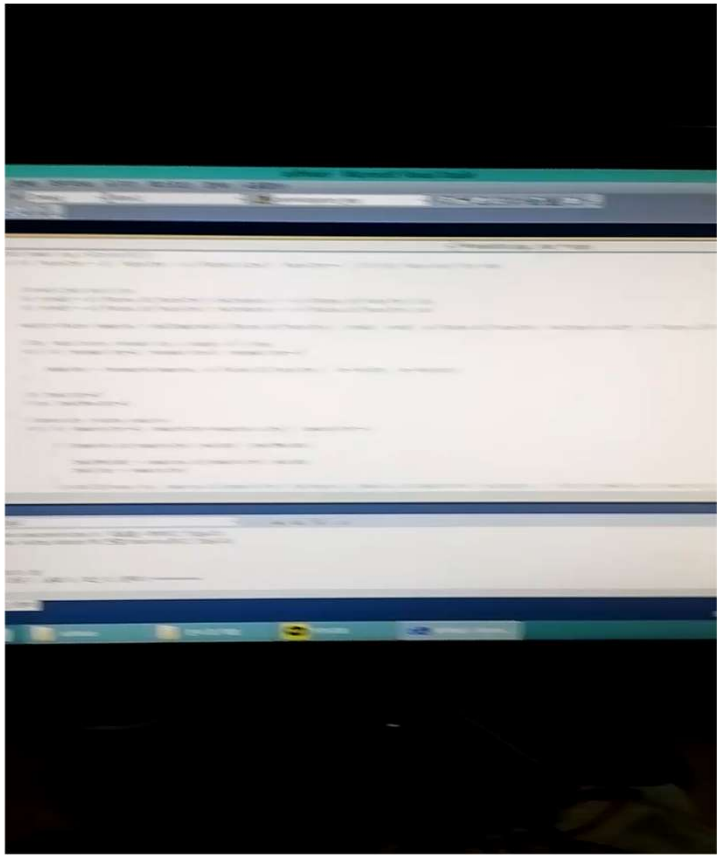
Repeat 1-3 until the best model is found with high confidence

Overview


We're at here
Mid-term exam
Project (maybe about 3DGS)
Project presentation
Final exam
Basic 3D concepts
Basic 3D applications (SfM and MVS)
Advanced 3D concepts
Advanced 3D applications
Do we really need 3D?
Recent 3D applications without 3D

First inspiring experience in computer vision

- Object tracking with particle filter
- Fourth-year undergraduate (2014)



Personal experiences with computer vision

- We can see the results — that makes it exciting
- Very practical and applicable engineering