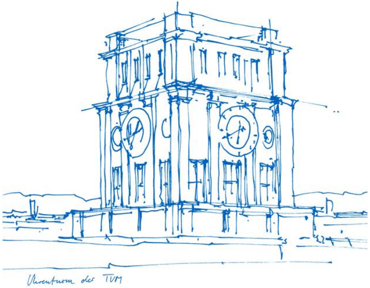
Computer Vision II: Multiple View Geometry (IN2228)

Chapter 10 Combination of Different Configurations

Dr. Haoang Li

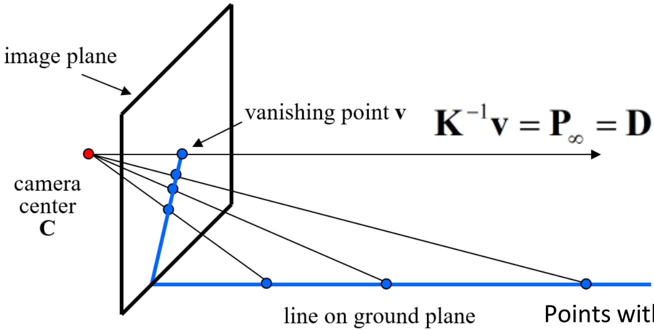
29 June 2023 11:00-11:45



Explanation for Point at Infinity

- Vanishing point and vanishing direction

✓ Vanishing point is NOT at infinity.
✓ Vanishing point is the projection of a 3D point at infinity.
✓ Vanishing direction is computed based on image normalization.



$$\mathbf{K}^{-1}\mathbf{v} = \mathbf{P}_\infty = \mathbf{D}$$

A point at infinity

Points with limited depths

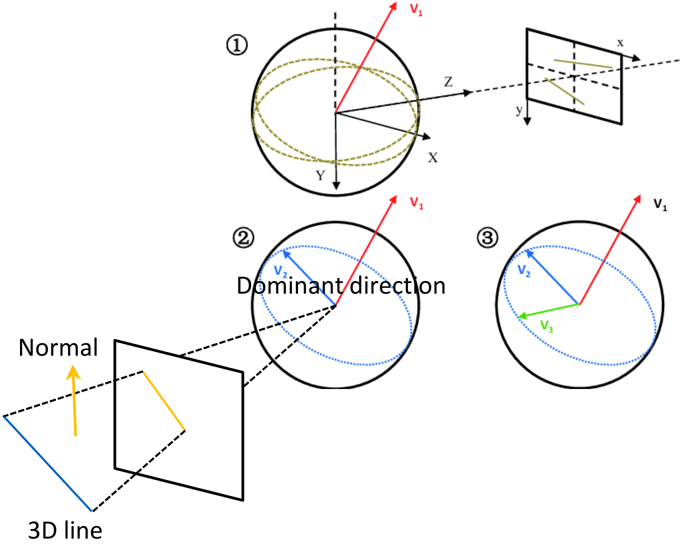
Explanation for Sampling-based Method

➢ Great Circle & Constraints of v2

✓ Great circle shown in blue is the intersection between the unit sphere and projection plane.

✓ Dominant direction v2 should be orthogonal to the normal of projection plane; should be parallel to the projection plane.

✓ v1 and v3 are irrelevant to the great circle shown in blue.



Today's Outline

- Knowledge Review
- Combination of 2D-2D and 3D-2D (Monocular Camera)
- Combination of 2D-2D and 3D-3D (Stereo Camera)
- Combination of 2D-2D and Single-view (Monocular Camera)

Knowledge Review

➢ 2D-2D Geometry

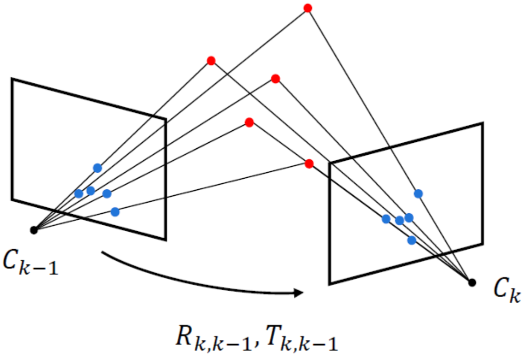
✓ Localization and mapping from 2D-to-2D point correspondences

Localization:

- 8-point algorithm
- 5-point algorithm (more popular)

Mapping:

- Triangulation

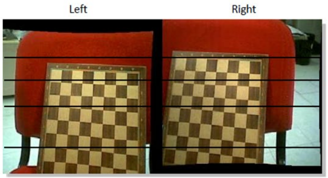


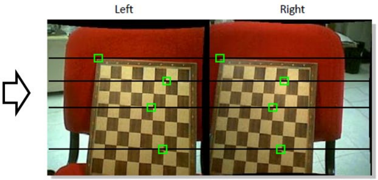
Knowledge Review

➢ 2D-2D Geometry

✓ Dense mapping from 2D-to-2D point correspondences

Dense correspondence establishment





Depth from disparity

$$Z_P = \frac{bf}{\boxed{u_l - u_r \quad \text{Disparity}}}$$

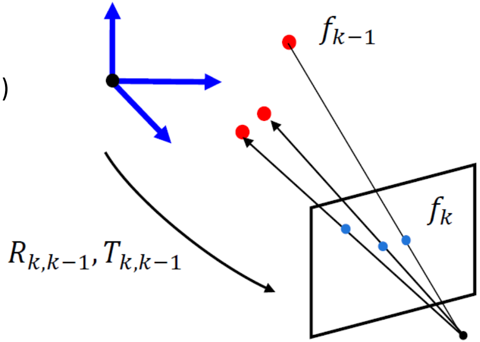
Knowledge Review

➢ 3D-2D Geometry

✓ Localization from 3D-to-2D point correspondences

Perspective-n-points (PnP) methods:

- DLT algorithm: minimal case: 6 points from 3D objects
- P3P algorithm: minimal case: 3 points (+1 for disambiguation)
- EPNP algorithm: for more than 4 points



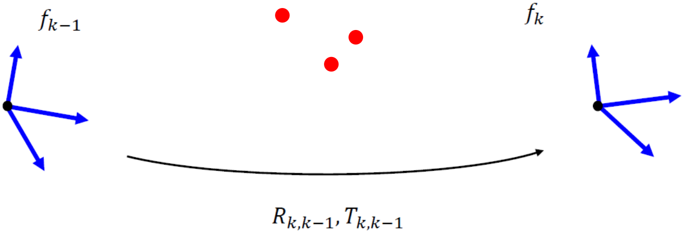
Knowledge Review

➢ 3D-3D Geometry

✓ Localization from 3D-to-3D point correspondences

Popular methods:

- Iterative method: ICP
- Non-iterative method: Closed-form solution based on SVD



Knowledge Review

➢ Single-view Geometry

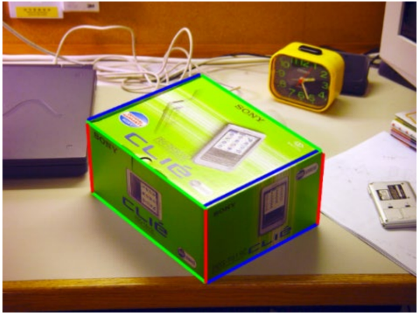
✓ Vanishing point estimation in a single image

Population method from a set of image lines:

- Census-based
- Sampling-based
- Search-based

Applications introduced before:

- Single-view calibration

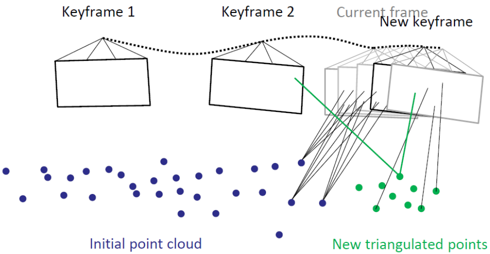


2D-2D and 3D-2D (Monocular Camera)

➢ Overview

✓ Alternately estimate camera pose and triangulate 3D points.

- Pose initialization based on 2D-2D correspondences (relative pose). Left camera frame is treated as the global world frame.
- Use the estimated camera pose and 2D-2D correspondences to triangulate initial 3D points.
- Use 3D-2D correspondences to estimate the camera pose of the current frame (absolute pose).

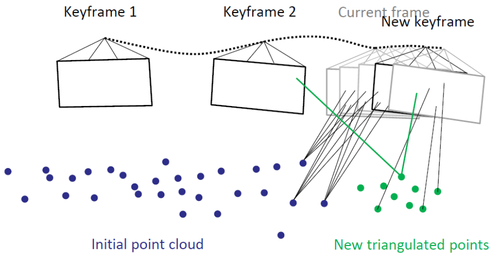


2D-2D and 3D-2D (Monocular Camera)

➢ Overview

✓ Alternately estimate camera pose and triangulate 3D points.

- Use the estimated **absolute** pose and new 2D-2D correspondences to triangulate new 3D points. 3D points are directly in the **world** frame.
- Use new 3D-2D correspondences to further estimate the pose of **new frame** (absolute pose).



2D-2D and 3D-2D (Monocular Camera)

➢ Detailed Procedures

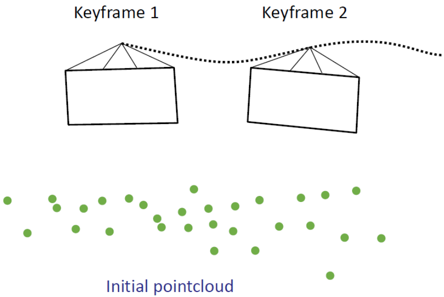
✓ First step

Initialization

- Pose initialization from two views: **5-point or 8-point RANSAC**

$$\boxed{t_1} = U \begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix} = \underbrace{\begin{bmatrix} U_0 & U_1 & U_2 \end{bmatrix}}_{\text{Columns}} \begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix} = \boxed{U_2 a}$$

Eigenvalue of essential matrix



- How far should two frames be?

We normalize the translation vector.

2D-2D and 3D-2D (Monocular Camera)

➢ Detailed Procedures

✓ First step

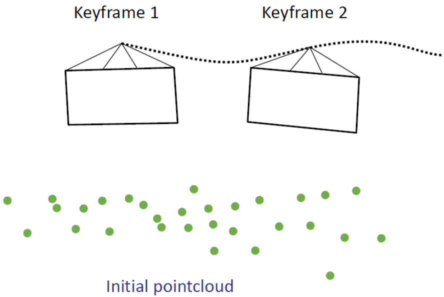
Initialization

- 3D point triangulation based on the estimated pose

$$\lambda_1 \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = K_1 [I|0] \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \qquad \lambda_2 \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} = K_2 [R|T] \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Goal: 3D points to triangulate

Estimated translation



- What scale is the point cloud?

It is determined by the norm of translation vector (the larger the norm is, the larger the point cloud is)

2D-2D and 3D-2D (Monocular Camera)

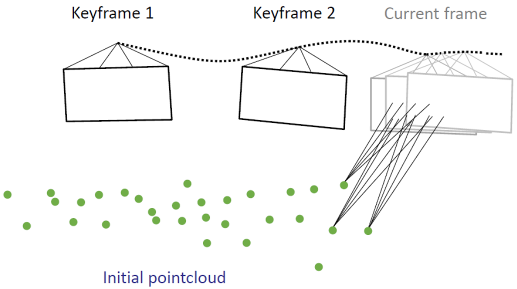
➢ Detailed Procedures

✓ Second step

Absolute pose estimation from 3D-2D point correspondences.

- Given a 3D point cloud (map) and associated 2D points, determine the absolute pose of new view.
- **Scale** of extrinsic parameters is aligned to the pre-defined scale of reconstructed 3D points.

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Known}} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}}_{\text{Unknown}} \cdot \underbrace{\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}}_{\text{Known}}$$



2D-2D and 3D-2D (Monocular Camera)

➢ Detailed Procedures

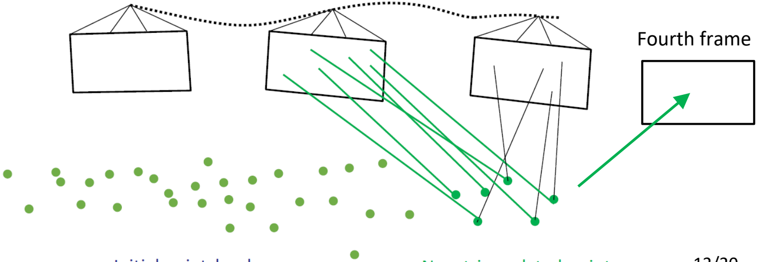
✓ Third step

Incremental 3D reconstruction and absolute pose estimation

- Use the estimated **absolute** camera pose to triangulate new 3D points in the **world** frame.
- Use 3D-2D correspondences to estimate the **absolute** pose of fourth frame.

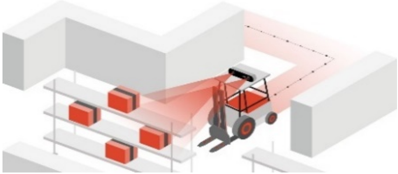
$$\lambda_2 \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} = K_2 [R|T] \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Absolute pose



2D-2D and 3D-3D (Stereo Camera)

➢ Overview



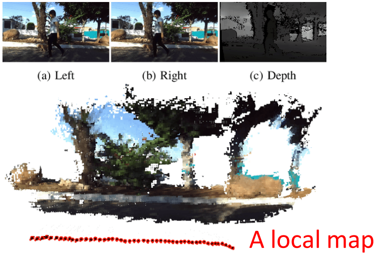
A global map

Alternately reconstruct 3D points and estimate camera pose.

- We have two images at a time. Estimate 3D points in the camera frame, i.e., **local map** (based on disparity of 2D-2D dense correspondences).

- Align this **local map** to the points in the world frame, i.e., **an incomplete global map** (based on 3D-3D correspondences) to estimate the **absolute** camera pose.

- Transform the reconstructed **local map** to the world frame to increment the **global 3D map**.



(d) 3D reconstruction

.

2D-2D and 3D-3D (Stereo Camera)

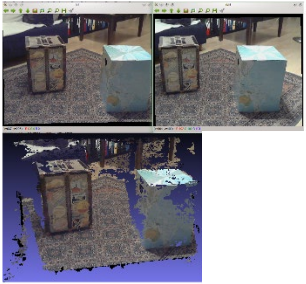
➢ Procedures

✓ Reconstruct 3D points in the left image

- We no longer need triangulation since we can directly obtain depth from disparity.

✓ Follow-up pose estimation of the left camera

- We aim to estimate the transformation from the **left camera frame** to the **world frame**.
- The motion between left and right camera frames are calibrated (measured) beforehand. So we can also obtain the trajectory of the **right** camera.



A local 3D map reconstructed by an image pair

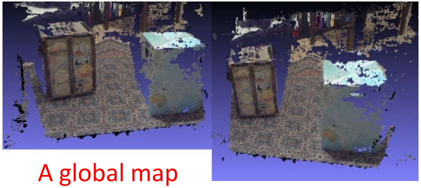
2D-2D and 3D-3D (Stereo Camera)

➢ Procedures

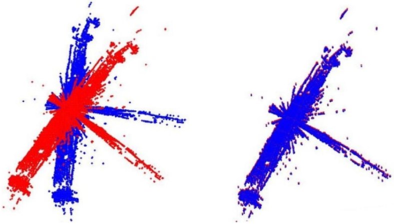
✓ Camera pose estimation by aligning point sets

Assume that we have reconstructed 3D point sets in both world frame (global) and left camera frame (local).

- We estimate the relative rotation and translation by aligning two point sets based on ICP.
- This practice is also common in Laser SLAM



Reconstructed global 3D map in the world frame and local 3D map in the left camera frame



Configuration of Laser SLAM

2D-2D and Single-view (Monocular Camera)

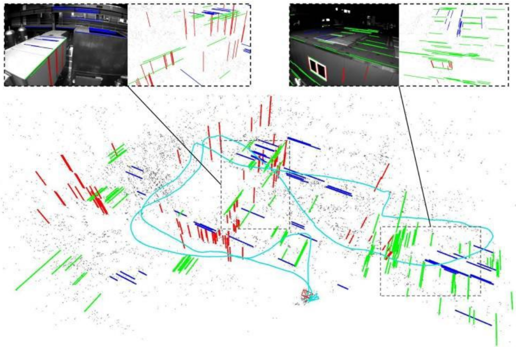
➢ Overview

✓ Application to monocular camera localization

- Camera Pose Estimation
- Camera Pose Optimization

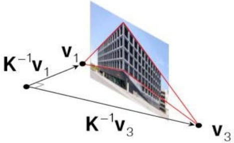
✓ Important clue

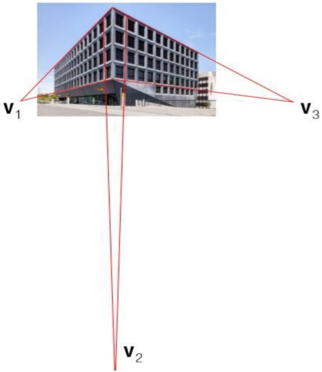
- Same dominant directions can be observed in two different images **without any overlap**.
- This constraint is a global constraint (traditional feature correspondences can only provide local constraint).



2D-2D and Single-view (Monocular Camera)

➢ Camera Pose Estimation

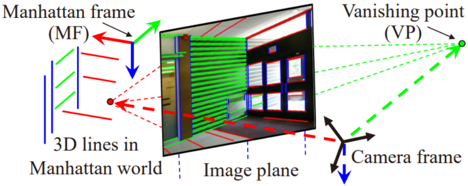




✓ Geometric Constraint

Computation of dominant directions in the camera frame

- Computing the vanishing directions starting from camera center based on **normalization** of vanishing points.
- The vanishing direction is aligned to a dominant direction. This constraint is introduced before.



2D-2D and Single-view (Monocular Camera)

➢ Camera Pose Estimation

✓ Geometric Constraint

Assume that we have obtained dominant directions in both *i*-th camera frame and *j*-th camera frame. (two image can have no overlap)

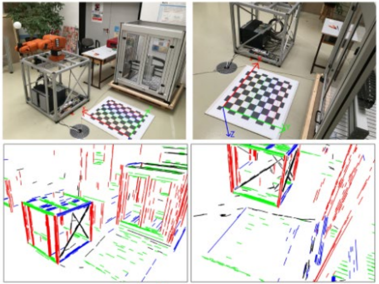
A relative rotation between two camera frames aligns these dominant directions.

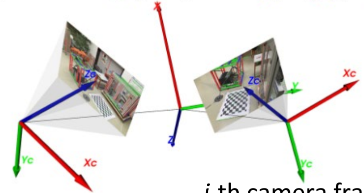
$$\boldsymbol{\delta}_k^j \propto \mathbf{R}_{ij} \boldsymbol{\delta}_k^i$$

Same k-th dominant directions in different camera frames

k is the ID of dominant direction

To estimate





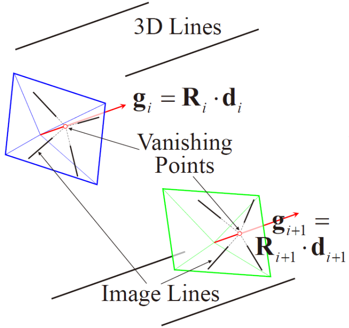
i-th camera frame

j-th camera frame

2D-2D and Single-view (Monocular Camera)

➢ Camera Pose Optimization

✓ Geometric Constraint



Here, we only consider a single dominant direction

Global ↑ Local ↑

$$\mathbf{g}_i = \mathbf{R}_i \mathbf{d}_i$$

i is the ID of camera

Rotation can be used to align the dominant directions from the **camera frame** to the **world frame**

$$[\hat{\mathbf{g}}_i^1, \hat{\mathbf{g}}_i^2, \cdots, \hat{\mathbf{g}}_i^N] = \mathbf{R}_i [\mathbf{d}_i^1, \mathbf{d}_i^2, \cdots, \mathbf{d}_i^N]$$

The dominant directions in the **camera frame** and the **world frame** are both known. We aim to find the optimal rotation that **best aligns** these dominant directions.

Summary

- Overview
- 2D-2D and 3D-2D (Monocular Camera)
- 2D-2D and 3D-3D (Stereo Camera)
- 2D-2D and Single-view (Monocular Camera)

Thank you for your listening!
If you have any questions, please come to me :-)