In this document, we highlight important knowledge in Chapters 06—10.
This will be highly relevant to the final exam.

Chapter 06 2D-2D Geometry (Part 1 Overview and Fundamentals)
Pages 07, 09, 14, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30

Chapter 06 2D-2D Geometry (Part 2 Camera Pose Estimation)
Pages 05, 08, 09, 10, 11, 12, 13, 14, 21, 22
Remark 1: For page 05, students are only required to understand the first two points, i.e., Basis of null space and the linear expression of vector e.
Remark 2: For page 08, students are not required to prove the lemma.
Remark 3: For pages 09—13, students are required to understand the derivation.

Chapter 06 2D-2D Geometry (Part 3 3D Reconstruction)
Pages 07, 08, 09, 10, 11, 12, 18, 19, 20, 21, 22, 23, 24, 35
Remark: For page 18, student are required to memorize the conclusion of Z_p computation.

Chapter 06 2D-2D Geometry (Part 4 Dense Correspondence Search and Homography)
Pages 03, 04, 08, 12, 13, 14, 15, 17, 18
Remark 1: For pages 13—14, students are required to understand how to derive Homography.
Remark 2: For page 17, students are not required to memorize the linear system.

Chapter 07 3D-2D Geometry
Pages 04, 06, 07, 08, 16, 17, 22, 31
Remark 1: For page 08, students are only required to memorize the conclusion.
Remark 2: For page 16, students are not required to memorize the linear system.

Chapter 08 3D-3D Geometry
Pages 04, 05, 06, 09, 10, 11, 12, 15, 16, 17
Remark 1: For page 09, students are only required to memorize the methods’ name.
Remark 2: For page 12, students are required to memorize the conclusion.

Chapter 09 Single-view Geometry
Pages 03, 04, 05, 11, 12, 13, 14, 15, 16, 17, 18, 21, 22, 30, 31
Remark 1: For page 03—05, students are only required to know the applications’ name.
Remark 2: For page 22, the search-based method will not be asked in the exam.

Chapter 10 Combination of Different Configurations
Pages 07, 08, 10, 11, 12, 13, 16, 18

06_2D-2D_01

P01 2D-2D : Camera pose estimation / 3D Reconstruction Traingulation

P07 Problem : Without 3D point → get K, R, T → use K, R, T, compute the 3D point
2D-2D

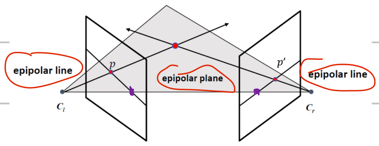
P09 : Give a set of n point correspondences : $\{ p_1^i = (u_1^i, v_1^i),\ p_2^i (u_2^i, v_2^i) \}$

Estimate camera pose R, t and 3D position P^i

$$\lambda_1^i \begin{bmatrix} u_1^i \\ v_1^i \\ 1 \end{bmatrix} = K_1 [I \mid 0] \begin{bmatrix} x_w^i \\ y_w^i \\ z_w^i \\ 1 \end{bmatrix} \qquad \lambda_2^i \begin{bmatrix} u_2^i \\ v_2^i \\ 1 \end{bmatrix} = K_2 [R \mid t] \begin{bmatrix} x_w^i \\ y_w^i \\ z_w^i \\ 1 \end{bmatrix}$$

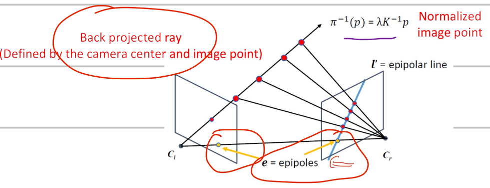
Loss scale = 2 DOF = up to scale / $n \geq 5$ Express in left camera

P14 Geometric constraints : Epipolar planes and lines :



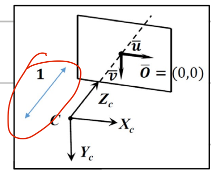
Camera center C_l C_r and image point p and p' ⇒ Epipolar plane

Intersection of Epipolar plane and image planes ⇒ Epipolar lines



Projection of the optical center on the other camera image ⇒ Epipole

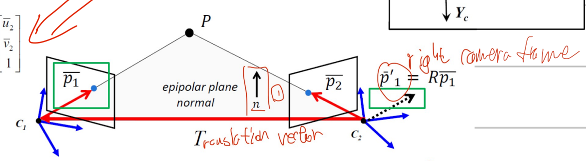
$\pi^{-1}(p) = \lambda K^{-1} p$



P16 : Essential Matrix : Coplanarity constraint

$\bar{p}_1 = \begin{bmatrix} \bar{u}_1 \\ \bar{v}_1 \\ 1 \end{bmatrix}$ $\bar{p}_2 = \begin{bmatrix} \bar{u}_2 \\ \bar{v}_2 \\ 1 \end{bmatrix}$

Normalized image coordinates



$$\begin{bmatrix} \bar{u}_1^i \\ \bar{v}_1^i \\ 1 \end{bmatrix} = K_1^{-1} \begin{bmatrix} u_1^i \\ v_1^i \\ 1 \end{bmatrix} \qquad \begin{bmatrix} \bar{u}_2^i \\ \bar{v}_2^i \\ 1 \end{bmatrix} = K_2^{-1} \begin{bmatrix} u_2^i \\ v_2^i \\ 1 \end{bmatrix}$$

Normalized image coordinates

$\bar{p}_1, \bar{p}_2, T$ **are coplanar**

Right camera frame Left camera frame

② $\underbrace{\boxed{\bar{p}_2^T \cdot n = 0}}_{\text{orthogonality}} \Rightarrow$ ③ $\bar{p}_2^T \cdot \underbrace{(T \times \bar{p}'_1)}_{\text{Normal of epipolar plane}} = 0 \Rightarrow \bar{p}_2^T (T \times (R \bar{p}_1)) = 0$ ⟺ $n = T \times (R \cdot \bar{p}_1)$

From dot product to matrix multiplication

16/32

$\bar{p}_2^T E \bar{p}_1 = 0$

$$\Rightarrow \begin{bmatrix} u_2^i \\ v_2^i \\ 1 \end{bmatrix}^T \underbrace{\boxed{K_2^{-T} E K_1^{-1}}}_{\parallel} \begin{bmatrix} u_1^i \\ v_1^i \\ 1 \end{bmatrix} = 0$$

F : Fundamental matrix

$\bar{p}_2^T (T \times R \bar{p}_1) = 0 \Rightarrow \bar{p}_2^T [T_\times] R \bar{p}_1 = 0 \Rightarrow \boxed{\bar{p}_2^T E \bar{p}_1 = 0}$

Essential matrix : $\boxed{E = [T_\times] R}$

P21 ① Eight-point method (DLT)

For n points, we can write need 8 correspondences

Normalized image coordinates $n \times 9$ 9×1

$$\underbrace{\begin{bmatrix} \bar{u}_2^1 \bar{u}_1^1 & \bar{u}_2^1 \bar{v}_1^1 & \bar{u}_2^1 & \bar{v}_2^1 \bar{u}_1^1 & \bar{v}_2^1 \bar{v}_1^1 & \bar{v}_2^1 & \bar{u}_1^1 & \bar{v}_1^1 & 1 \\ \bar{u}_2^2 \bar{u}_1^2 & \bar{u}_2^2 \bar{v}_1^2 & \bar{u}_2^2 & \bar{v}_2^2 \bar{u}_1^2 & \bar{v}_2^2 \bar{v}_1^2 & \bar{v}_2^2 & \bar{u}_1^2 & \bar{v}_1^2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \bar{u}_2^n \bar{u}_1^n & \bar{u}_2^n \bar{v}_1^n & \bar{u}_2^n & \bar{v}_2^n \bar{u}_1^n & \bar{v}_2^n \bar{v}_1^n & \bar{v}_2^n & \bar{u}_1^n & \bar{v}_1^n & 1 \end{bmatrix}}_{Q \text{ (this matrix is known)}} \underbrace{\begin{bmatrix} e_{11} \\ e_{12} \\ e_{13} \\ e_{21} \\ e_{22} \\ e_{23} \\ e_{31} \\ e_{32} \\ e_{33} \end{bmatrix}}_{\bar{E} \text{ (this matrix is unknown)}} = 0 \quad \Rightarrow \quad Q \cdot \bar{E} = 0$$

Minimum solution : $n = 8$ / rank(Q) = 8

Overdetermined solution : $n > 8$

minimize $\| Q\bar{E} \|$, s.t. $\|\bar{E}\|^2 = 1$

Smallest eigenvalue ⟺ eigen vector

SVD($Q^T Q$)

P_{25}

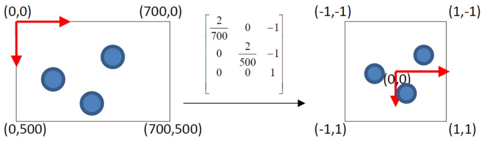
$$\begin{bmatrix} u_2^i \\ v_2^i \\ 1 \end{bmatrix}^T F \begin{bmatrix} u_1^i \\ v_1^i \\ 1 \end{bmatrix} = 0 \Rightarrow \begin{bmatrix} u_2^1 u_1^1 & u_2^1 v_1^1 & u_2^1 & v_2^1 u_1^1 & v_2^1 v_1^1 & v_2^1 & u_1^1 & v_1^1 & 1 \\ u_2^2 u_1^2 & u_2^2 v_1^2 & u_2^2 & v_2^2 u_1^2 & v_2^2 v_1^2 & v_2^2 & u_1^2 & v_1^2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_2^n u_1^n & u_2^n v_1^n & u_2^n & v_2^n u_1^n & v_2^n v_1^n & v_2^n & u_1^n & v_1^n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

Similiary, use F to solve $\|QF\|$

problem: magnitude difference between Q

$\rightarrow$ poor LS - result

solution: normalized - 8 - points



Centroid of each set is 0

mean standard deviation $\sqrt{2}$

$$\hat{p}^i = \frac{\sqrt{2}}{6}(p^i - \mu)$$

$$\hat{p}^i = \underbrace{\begin{bmatrix} \frac{\sqrt{2}}{\sigma} & 0 & -\frac{\sqrt{2}}{\sigma}\mu_x \\ 0 & \frac{\sqrt{2}}{\sigma} & -\frac{\sqrt{2}}{\sigma}\mu_y \\ 0 & 0 & 1 \end{bmatrix}}_{B} p^i$$

$$\mu = (\mu_x, \mu_y) = \frac{1}{N}\sum_{i=1}^{n} p^i$$

$$6^2 = \frac{1}{N}\sum_{i=1}^{n} \| p^i - \mu \|^2$$

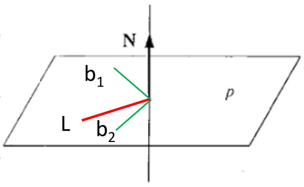
① $\hat{P}_1 = B_1 P_1$ / $\hat{P}_2 = B_2 P_2$

② $\hat{P}_2^T \hat{F} \hat{P}_1 = 0 \Rightarrow \hat{F}$

③ $\underbrace{P_2^T B_2^T \hat{F} B_1}_{=F} P_1 = 0$

06 _ 2D_2D _ 02

P02 Null space and Rank



basis of null space $\{b_1, b_2\}$ / Num(unknown) = Rank(N) + Dim(Null space)

$N^T L = 0$

P05 Five-point method: Basis of null space: X, Y, Z, W Dim(null space) = 4

know 9D basis vector

$e = xX + yY + zZ + wW$

x, y, z are unknown coefficients, $w = 1$

P08 Essential matrix decomposition:

$$E = U\Sigma V^T = U\boxed{\begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & 0 \end{bmatrix}}V^T = [\mathbf{t}]^\wedge R$$

Mathematical form

Geometric form

$\Sigma = [6, 6, 0]^T$

$$\boxed{[\mathbf{t}_1]^\wedge = U\begin{bmatrix} 0 & a & 0 \\ -a & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}U^T} = [U\begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix}]^\wedge$$

3*1 vector

Matrix

$$\Rightarrow \boxed{\mathbf{t}_1} = U\begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix} = [U_0 \quad U_1 \quad U_2]\begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix} = \boxed{U_2 a}$$

Columns

Translation

$t_1 = U_2 a$

$\Rightarrow$ ~~$t_1 = U$ 第三列~~

$$\boxed{R_1 = U\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}V^T}$$

Rotation

SVD of E

Decomposition of Σ

$$E = U\boxed{\begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & 0 \end{bmatrix}}V^T = U\boxed{\begin{bmatrix} 0 & a & 0 \\ -a & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}V^T$$

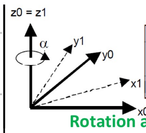
$$= \boxed{U\begin{bmatrix} 0 & a & 0 \\ -a & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}U^T}\boxed{U\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}V^T} = [\mathbf{t}]^\wedge R$$

Skew-symmetric matrix

Associative law

Introducing an **identity** matrix for derivation

P12 Another Σ



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

Rotation along the z-axis (introduced before)

$$\Sigma = \begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & 0 \end{bmatrix} = \boxed{\begin{bmatrix} 0 & -a & 0 \\ a & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}$$

$$\boxed{\mathbf{t}_2} = U\begin{bmatrix} 0 \\ 0 \\ -a \end{bmatrix} = -U_2 a = \boxed{-\mathbf{t}_1}$$

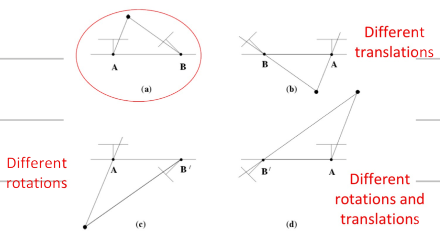
Another rotation and translation results

$$R_2 = U\boxed{\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}V^T \triangleq UW^TV^T$$

$R_1 = UR_z(+\frac{\pi}{2})V^T, [\mathbf{t}_1]^\wedge = UR_z(+\frac{\pi}{2})\Sigma U^T$

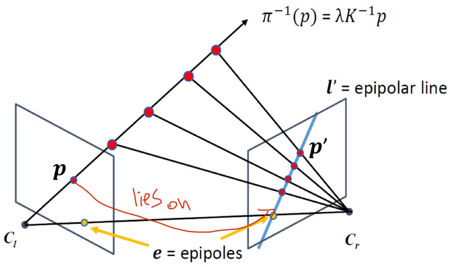
$R_2 = UR_z(-\frac{\pi}{2})V^T, [\mathbf{t}_2]^\wedge = UR_z(-\frac{\pi}{2})\Sigma U^T$

↑ at last Four solutions



$\Leftarrow$ use 3D reconstruction point to fill out

P21 1-D correspondence : 2D point correspondence in two images are exhaustive computationally



potential matches for P must lie on the epipolar liner L'
⇒ epipolar constraint

constraint the search problem into 1-D search along L'

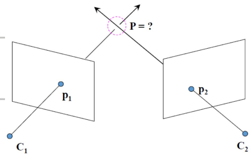
All epipolar liner intersect at epipole

06 _ 2D - 2D _ 3

P07 Triangulation : Determining the 3D position of a point given the 2D corresponding points and KRT

P08 Want to intersect two rays of p_1 and p_2 at P

↳ but noise and numerical error → only get approximation



Left camera: $\lambda_1 \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = K_1 \boxed{[I|0]} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$

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

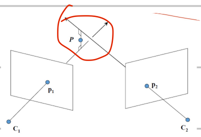
⇐ express $P = \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix}$ in left

Left camera: $\lambda_1 \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = \boxed{K_1[I|0]} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \Rightarrow \lambda_1 p_1 = \boxed{M_1} \cdot P \Rightarrow 0 = p_1 \times M_1 \cdot P \Rightarrow [p_1 \times] \cdot M_1 \boxed{P} = 0$

Known | Collinearity (up-to-scale) | Cross product | Unknown

Right camera: $\lambda_2 \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} = \boxed{K_2[R|T]} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \Rightarrow \lambda_2 p_2 = \boxed{M_2} \cdot P \Rightarrow 0 = p_2 \times M_2 \cdot P \Rightarrow [p_2 \times] \cdot M_2 \cdot \boxed{P} = 0$

SVD get P



⇐ Geometrically, P is computed as the midpoint of the shortest 3D line segment

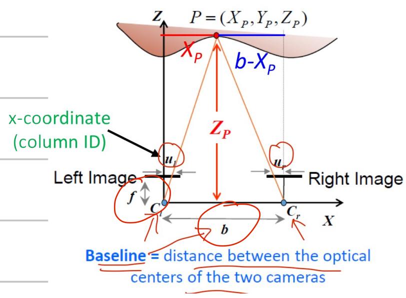
↳ Least Square Approximation

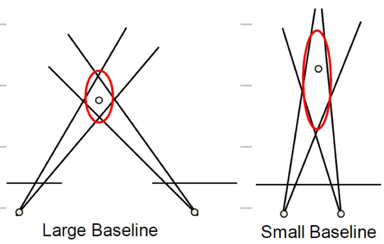
P18 Stereo Vision : Compute disparity and depth

$$Z_P = \frac{bf}{u_l - u_r}$$

Baseline, Focal length, Depth, Disparity

(After Rectification)





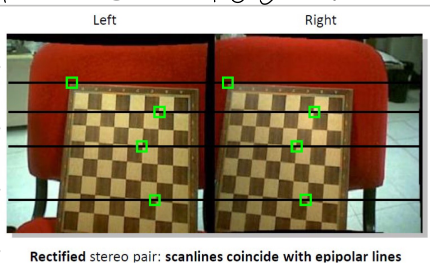
optimal baseline b?

Large baseline : small depth error, difficult search problem (object disappear)

Small baseline : Large depth error, easy search problem

P20 Stereo Rectification : best ⇒ epipolar lines are aligned to the horizontal scanlines

↳ 1D search ⇒ along the same scanlines



Rectified stereo pair: scanlines coincide with epipolar lines

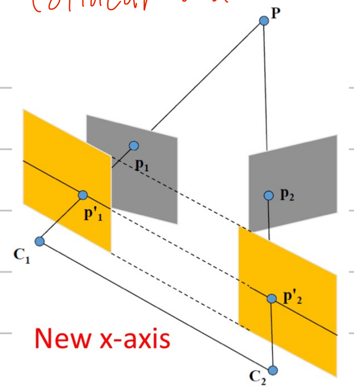
Method :

Wrap two image planes parallel to the baseline

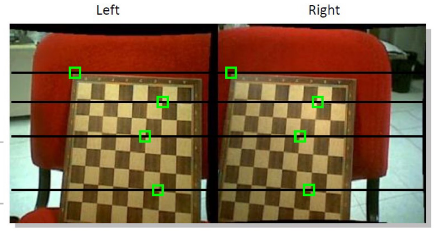
↳ new epipolar lines are aligned to the horizontal scanlines

↳ ≈ 2D Transformation ⇒ new 2D projections

Colinear and horizontal



Same y ⇒



Rectified stereo pair: scanlines coincide with epipolar lines

Define two rotate matrixs

↳ Base line must parallel to new X-axis ⇒ horizontal

↳ All corresponding points must have same y-coordinate (row ID)

⇒ have same intrinsic parameters

⇒ 2D Transformation caused by extrinsic matrix (R|T)

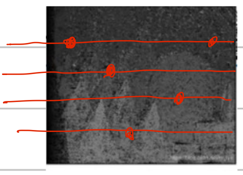
P35 Disparity Map → 3D point cloud :

$$\begin{cases} z = depth(i, j) \\ x = \dfrac{(j - c_x) \times z}{f_x} \\ y = \dfrac{(i - c_y) \times z}{f_y} \end{cases}$$

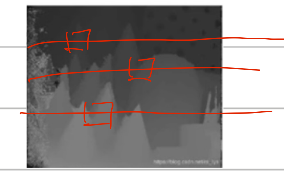
$$Z_p = \frac{bf}{u_l - u_r}$$

06 _ 2D-2D _ 04

P03 Dense Correspondence : 1-D search along the scan line
↳ pixel wise similiarity
↳ block wise similicarity

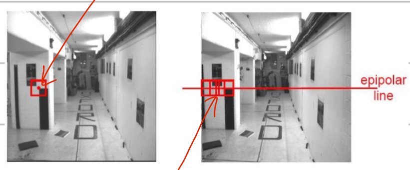


Disparity result based on **pixel-wise** similarity



Disparity result based on **block-wise** similarity

Window around pixel



block/patch along the line

Scale and viewpoint arn't changed for Stereo camera

Use a window around pixel to average the noise

minimize NCC, SSD, ...

P08 Effects of window sizes (w) : smaller window : more detail / more noise

larger window : smoother disparity map / less detail

P12 Homography : A transformation of point correspondences 2D-2D

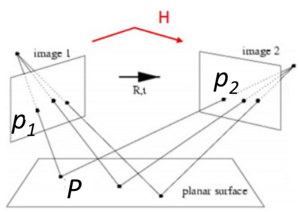
↳ encodes the co-planarity information

Method : 3D point expression : $n^T P + d = 0 \Rightarrow -\frac{n^T P}{d} = 1$ (3D) ← left camera frame $P_1 = k_1 P$

projective geometry : $P_2 = k_2[R|T]\,P = k_2\,(RP + t)$

$= k_2\,(R \cdot P - t \cdot \frac{n^T P}{d}) = k_2\,(R - t \cdot \frac{n^T}{d}) \cdot P$

$= \underbrace{k_2\,(R - t \cdot \frac{n^T}{d}) \cdot k_1^{-1}}_{H} P_1$



3D point P is projected to both left and right image planes

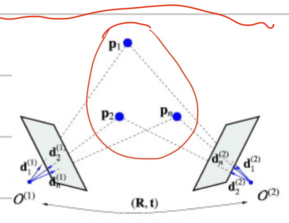
$\Rightarrow P_2 = H P_1 \qquad H = k_2\,(R - t \cdot \frac{n^T}{d}) \cdot k_1^{-1}$

P15 Application: A pair of corresponding points $q_2 \propto H q_1$

$$\begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} \propto \begin{pmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} & \mathbf{H}_{13} \\ \mathbf{H}_{21} & \mathbf{H}_{22} & \mathbf{H}_{23} \\ \mathbf{H}_{31} & \mathbf{H}_{32} & \underbrace{\boxed{\mathbf{H}_{33}}}_{=1} \end{pmatrix} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix}$$

⇐ Homography is up to Scale has 8 DoF

Need four point correspondences $QH = (\cdot)$

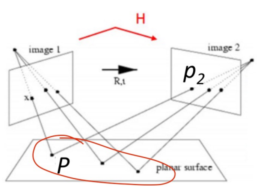


Arbitrary 3D points

P18 Essential Matrix vs Homography :

Ess... : Arbitrary 3D points 5 or 8 or more correspondence

Homo : coplanar 3D points only need 4 correspondence



3D points lying on the same 3D plane

07_3D_2D

P03 ~~Pose~~ : Relative Pose : pose from right camera frame to left camera frame

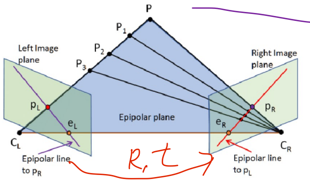
Absolute Pose : pose from camera frame to world frame

P04 2D2D vs. 3D2D : 2D2D → Relative camera pose estimation

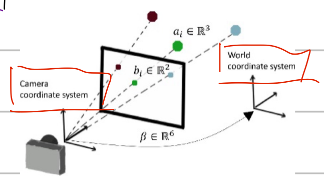
Not suitable to estimate absolute pose of sequential images

↳ Time-consuming / Translation up to scale

3D2D → Absolute camera pose estimation



Relative camera pose



Absolute camera pose

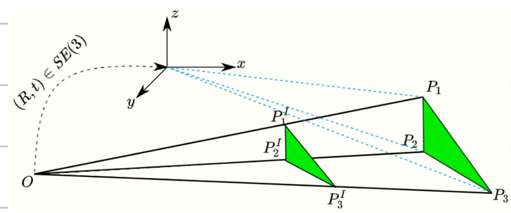
different from calibration ⇒ K/RT

unknown

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K\boxed{RT} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

P08 PnP : Give Point Correspondences and K ⇒ $\boxed{RT}$

- 2 points : infinite number of solutions
- 3 points : Minimal Case
- 4 points : more reliable



P16 PnP Algorithm (DLT) :

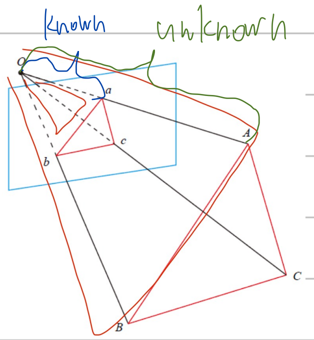
$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K[R|T] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} t_{11} & t_{12} & \\ & \ddots & \\ & & \end{bmatrix}}_{[R|T]} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} t_1^T \\ t_2^T \\ t_3^T \end{bmatrix} P$$

$$\Rightarrow \begin{cases} t_1^T P - \overbrace{\boxed{t_3^T P}}^{s} u = 0 \\ t_2^T P - \boxed{t_3^T P} v = 0 \end{cases} \Rightarrow \begin{pmatrix} P_1^T & 0 & -u_1 P_1^T \\ 0 & P_1^T & -v_1 P_1^T \\ \vdots & \vdots & \vdots \\ 0 & P_n^T & -v_n P_n^T \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} = 0$$

dim = 12

⇒ need 6 point correspondence

P17 P3P :



Configuration of P3P problem

$$\begin{cases} \Delta Oab - \Delta OAB \\ \Delta Obc - \Delta OBC \\ \Delta Oac - \Delta OAC \end{cases}$$

Three pairs of triangles ⇐ 3 point

P_{22} EPnP : Express each 3D point by a linear combination of four control points

6 point

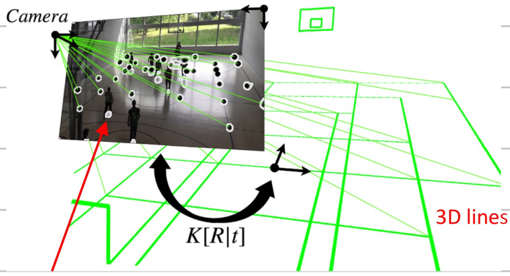
Two-step : ① detemine the coordinates of four control points in camera and world frame

② use control point to obtain the coordinates of 3D points in camera and world frame

③ base 3D-3D correspondences to compute [R|T]

P_{31} PnL : Input : a set of 3D-2D Line correspondence / 3D lines in world frame

Output : 3-DoF Rotation and 3-DoF Translation aligning the world frame to camera frame



Endpoints of image line segments

08 _ 3D - 3D

P04 3D-3D : Input : Two point sets f_{k-1} and f_k in 3D

Minimal case : Three 3D-3D point correspondences

solving :

$$\begin{bmatrix} X^i_{k-1} \\ Y^i_{k-1} \\ Z^i_{k-1} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \cdot \begin{bmatrix} X^i_k \\ Y^i_k \\ Z^i_k \\ 1 \end{bmatrix} \Rightarrow \text{solve } [R|t]$$

where i is the feature ID. k is the point sets

P05

Input : $X = \{x_1, \cdots, x_{N_x}\}$ Number of points are unnecessarily same

$P = \{p_1, \cdots, p_{N_p}\}$ Don't know which points are corresponding

Goal : Find optimal $[R|t]$ minimizes the sum of squard error

$$E(R,t) = \frac{1}{N_p} \sum_{i=1}^{N_p} ||x_i - R p_i - t||^2$$

Unknown

Point to transform

⇐ x_i and p_i are unknown-but-sought corresponding

P06 ① If point correspondences are known, closed-form solution RT (non-iterative)

② If point correspondences are unknown, perform iterations

P09 Non-iterative Method : SE(3) : ...

SIM(3) : Horn's / Umeyama's

SE(3) Method : ① compute center of mass

$\mu_x = \frac{1}{N_x} \sum_i^{N_x} x_i$ $\mu_p = \frac{1}{N_p} \sum_i^{N_p} p_i$ ⇐ assume $N_x = N_p$

② point sets normalizations

$X' = \{x_i - \mu_x\} = \{x_i'\}$ $P' = \{p_i - \mu_p\} = \{p_i'\}$

③ Compute Matrix W

$W = \sum_{i=1}^{N_p} \boxed{x_i' p_i'^T}$

④ SVD on W

$$W = U \begin{bmatrix} \sigma_1 & & 0 \\ & \sigma_2 & \\ 0 & & \sigma_3 \end{bmatrix} V^T$$

⑤ optimal solution

$R = UV^T$ / $t = \mu_x - R\mu_p$

P15 Iterative closest Point (ICP): Idea: Iterative align two point sets ⇒ "close enough"

Method: ① choose a pair of points with smallest distance

② compute R, t based on SVD

③ Apply R, t to the set of points

④ compute $E(R,t)$

⑤ If error decrease and error > threshold

Yes

no

09_ Single View

P03 Applications of single view :

Single-view 3D reconstruction : Vanishing points and 3D box priors

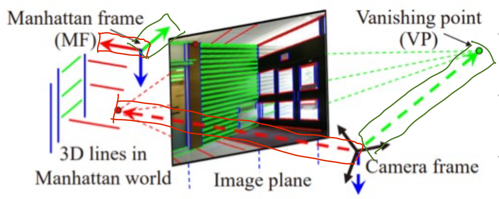
Camera pose estimation and optimization : Vanishing point ⇒ rotation

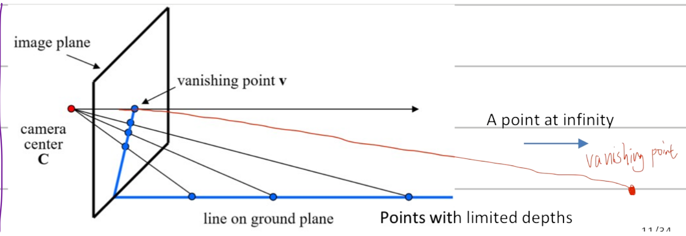
Camera calibration : Intrinsic parameters

P09 Vanishing point : 2D lines projected from parallel 3D lines intersect at a vanishing point

Vanishing point and camera center ⇒ vanishing direction

Vanishing direction is parallel to a 3D dominant direction





P11 Expression : Vanishing point v = projection of a point at infinity

Ray from camera center C through v is parallel to 3D line

Two parallel 3D lines have the same vanishing point v

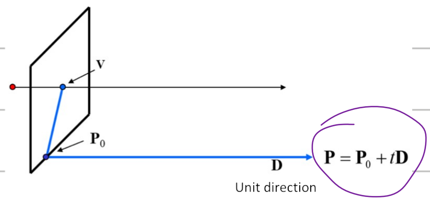
An image may have many vanishing points

P13 Mathematical : $P = P_0 + tD$ / $P_t = \begin{bmatrix} P_x + tD_x \\ P_y + tD_y \\ P_z + tD_z \\ 1 \end{bmatrix} = \begin{bmatrix} P_x/t + D_x \\ P_y/t + D_y \\ P_z/t + D_z \\ 1/t \end{bmatrix} \Rightarrow P_\infty \overset{t=\infty}{=} \begin{bmatrix} D_x \\ D_y \\ D_z \\ 0 \end{bmatrix}$

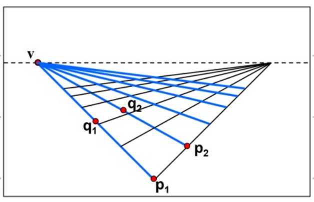
up to scale

↑ 3D point

↓ 2D point



up to scale ⟶ $\boxed{\begin{aligned} v &= KP_\infty = KD \\ K^{-1}v &= P_\infty = D \end{aligned}}$



compute v based on 2D lines : $v = (p_1 \times q_1) \times (p_2 \times q_2)$

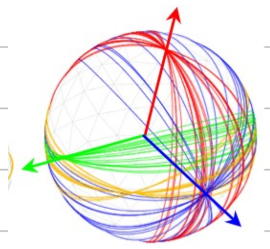
better use more lines

P16 Representation on Sphere : if image lines are parallel ⇒ vanishing points far from c

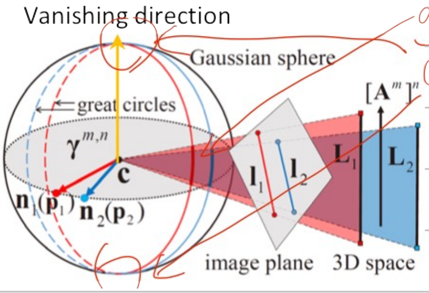
unreliable

① Project line onto a sphere ⇒ arc

② compute intersection of arc ⇒ vanishing direction



Vanishing directions



get these arc based on projection

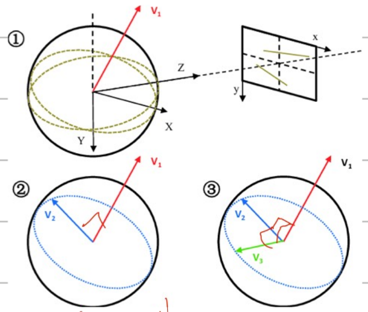
compute the Intersection

vanishing parallel to 3D lines direction

plane ⇒ infinite / no constraint

vanishing directions / sphere ⇒ orthogonality constraint

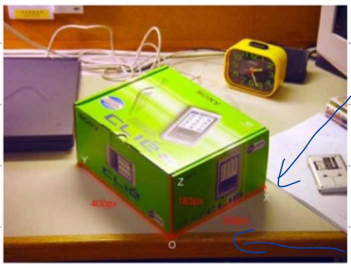
P21 Three dominant Strategies : ① use two circles determine first V_1
② use one circle determine second V_2 } three circles
↳ in third circle / orthogonal to V_1
③ based V_1 and V_2 determine V_3
↳ $V_3 \perp V_1$ / $V_3 \perp V_2$



orthogonal

* Multiple sampling ⇒ guarantee valid sampling

P30 Camera Calibration :



$P_{x\infty} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

$P_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$

$\Pi = K[R|t] = [\pi_1 \ \pi_2 \ \pi_3 \ \pi_4]$

3D point at infinity

$V_x = \Pi [1 \ 0 \ 0 \ 0]^T = \pi_1$

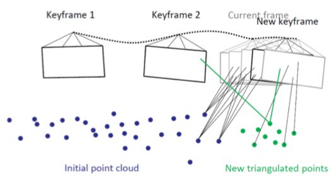
$V_y = \pi_2$

$V_z = \pi_3$ unknown ⇒ $\Pi = [V_x \ V_y \ V_z \ 0]$ up-to-scale

$O = \pi[0 \ 0 \ 0 \ 1]^T = \pi_4$

10 _ Combination   Left camera frame = Global world frame

P07 2D-2D and 3D-2D (Monocular Camera): Alternately estimate camera pose / triangulate 3D



① 2D-2D ⇒ Relative Pose RT
② Triangulate initial 3D points P
③ 3D-2D ⇒ Absolute Pose
④ 2D-2D / Absolute Pose ⇒ 3D Points in world frame
⑤ 3D-2D ⇒ New frame (Absolute Pose)

scale is $\|t\|_2$

P09 Detail: ① 2D-2D: 8/5 points ⇒ $[R|t]$ $\|t\|_2 = 1$

② get 3D points ⇒ $\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K_1 \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$ $\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K_2 [R|t] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$

③ Absolute pose of new frame ⇒

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K [R|t] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

④ get new 3D points ⇒ $\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K [R|t] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$

3D-2D get new frame absolute pose $[R|t]$

P13 2D-2D and 3D-3D (Stereo Camera):

① 2 input images (Stereo Camera) ⇒ Dense correspondences ⇒ Local Map (3D point)
② Align local map with 3D points in world frame ⇒ incomplete global map and absolute pose
③ Transform local map increment the global 3D map

P16 2D-2D and Single-view (Monocular Camera)

⇒ Camera pose Estimation / Optimization
⇒ Same dominant directions = two different images without any overlap
⇒ Global constraints

P18 Camera pose estimation: assume get dominant directions in both camera frame

relative rotation ⇒ $\delta_k^j \propto \boxed{R_{ij}} \delta_k^i$ ⇐ domain direction

between camera frame   estimate