In this document, we highlight important knowledge in Chapters 00—05. 期末不会涉及首页
This will be highly relevant to the final exam.

Chapter01_Mathematical_Backgrounds

Pages 02, 04, 05, 09, 10, 12, 16, 20, 22, 25, 27

Chapter 02 Motion and Scene Representation (Part 1 Basic Expression)

Pages 08, 12, 15, 16, 17, 18, 20, 24, 27, 28, 31, 43, 48

Chapter 02 Motion and Scene Representation (Part 2 Lie Group and Lie Algebra)

Pages 04, 05, 09, 10, 11, 12

Remark: Additional important knowledge will be introduced in the future. Students are required to understand the application of Lie Group and Lie Algebra to bundle adjustment.

Chapter 03 Image Formation (Part 1 Perspective Projection)

Pages 07, 09, 10, 15, 17, 18, 19, 20, 21, 22, 26, 27, 31, 33, 34, 36, 44, 50

Remark: Detailed information about Pages 09 and 10 will be introduced in the future.

Chapter 03 Image Formation (Part 2 Distortion and Supplementary Knowledge)

Pages 02, 04, 05, 14, 16, 17, 25

Chapter 04 Camera Calibration

Pages 02, 03, 07, 08, 14, 15, 16, 18, 19, 20, 21, 27, 28, 29, 30, 31, 32, 33, 34, 35, 39, 40, 41

Remark 1: For page 18, students are not required to know details about QR decomposition. Exam will not involve mathematical computation related to QR decomposition.

Remark 2: For pages 33, 34, and 35, students are required to understand the main idea of camera parameter recovery. Exam will not involve specific derivations.

Chapter 05 Correspondence Estimation (Part 1 Small Motion)

Pages 04, 05, 06, 07, 12, 16, 17, 18, 19, 20, 22, 23

Remark 1: For pages 16, 17, and 18, students are required to know how to derive M-matrix.

Remark 2: For pages 19 and 20, exam will not involve the computation of eigenvalues and eigenvectors.

Chapter 05 Correspondence Estimation (Part 2 Large Motion)

Pages 05, 08, 09, 10, 12, 14, 15, 16, 17, 20

Remark: For pages 09 and 10, students are required to know how to derive M-matrix.

Chapter 05 Correspondence Estimation (Part 3 Clarification and SIFT)

Page 03, 07, 12, 13, 14, 15, 16, 17, 18, 20, 24, 26, 28, 29, 31, 32, 33, 35, 38, 42, 43

01_Math

P02 Dot Product : $a \cdot b = a_1 \cdot b_1 + \cdots + a_n \cdot b_n$ / $a \cdot b = \|a\| \|b\| \cdot \cos\theta$ / how similiar / ~~$proj_a b = \frac{a \cdot b}{\|a\|} \cdot \frac{a}{\|a\|}$~~

P04 Cross Product : $a \times b = \cdots$ / orthogonal to both a and b / $a \times b = \|a\| \cdot \|b\| \cdot \sin\theta = S$

~~P08 Kronecker Product : $A \otimes B$~~

linearly independent / also exactly 3 vector

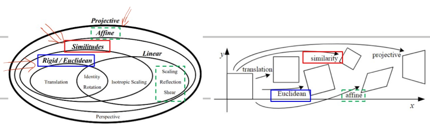
P09 Basis B in vector space V : $B = \{v_1, v_2, \cdots v_m\}$ $\quad V = a_1 \cdot v_1 + \cdots a_m \cdot v_m$

$m = R^m \quad R^3$ at least 3 vector

P10 Linear Span : for $v_1, v_2, \cdots v_m \in V$, $span(v_1, \cdots v_m)$ contain all the linear combinations

$\Rightarrow V = a_1 \cdot v_1 + \cdots + a_m \cdot v_m$

Overview



P12 Linear Independance : $x_1 \cdot v_1 + \cdots + x_k \cdot v_k = 0 \iff$ cofficiency $= 0$

~~P14~~ GS-orth : ortho-normalizing : $(v_1, v_2, \cdots v_m) \Rightarrow (u_1, u_2, \cdots u_m) \Rightarrow \left(\frac{u_1}{\|u_1\|}, \frac{u_2}{\|u_2\|}, \cdots, \frac{u_m}{\|u_m\|}\right)$

~~P18~~ Affine Transformation = Linear Transformation + Translation Transformation

P20 Rank = # linearly independent column = # non zero column

P21 Trace $= \sum_{i=1}^{N} a_{ii} = a_{11} + a_{22} + \cdots + a_{nn}$

inversable

P22 det = 2D area / 3D volume / $\det \neq 0 \iff$ unique solution // $\det = 0 \iff$ infinit solution

P25 kernel : $N(A) = Null(A) = ker(A) = \boxed{\{x \in k^n \mid Ax = 0\}}$ / Non trival solution $x \neq 0$.

P26 Skew-symmetric : $A = -A^T$ / $a = (a_1\ a_2\ a_3)^T$ $\quad A = [a]_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$

P27 Eigenvector/value : $Av = \lambda v$ / $(A - \lambda I)v = 0$ / $|A - \lambda I| = 0$

P30 Matrix Inverse : $AB = BA = I_N$ / $A^{-1} = Q \Lambda^{-1} Q^{-1}$ / $\det(A) \neq 0$

P31 SVD : $A_{m \times n} = U_{m \times m} \Sigma_{m \times n} V^T_{n \times n}$ / Σ is eigenvalue of $A^T A$

P32 QR : $A = QR$ / Q : orthogonal matrix , R : upper triangular matrix / $Q^T = Q^{-1}$

02_ Motion and Scene Representation

P08 Rigid Transformation : $x_w = R x_r + t$ / $R = \begin{bmatrix} c\theta & -s\theta \\ s\theta & c\theta \end{bmatrix}$ $t = [t_x \ \ t_y]^T$

P12 $a = Ra'$ / $\underline{SO(n)} = \{ R \in R^{n\times n} \mid RR^T = I, \det(R) = 1 \}$ / $a' = R^{-1} a = R^T a$ // $a' = Ra + t$

Special ortogonal Group

P14 $\begin{bmatrix} a' \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ 1 \end{bmatrix} \triangleq T \begin{bmatrix} a \\ 1 \end{bmatrix}$ // $\underline{SE(3)} = \{ T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \in R^{4\times 4} \mid R \in SO(3), t \in R^3 \}$

Special Euclidean Group

P15 $Y = RX + t \Rightarrow X = R^T(Y - t) \Rightarrow X = R^T Y - R^T t \Rightarrow T^{-1} = \begin{bmatrix} R^T & -R^T t \\ 0 & 1 \end{bmatrix}$

P16 Absolute → Relative Point : $X_w = R_1 X_1 + t_1 = R_2 X_2 + t_2 \Rightarrow \underbrace{R_2^T R_1}_{R_{12}} X_1 + R_2^T \underbrace{(t_1 - t_2)}_{t_{12}} = X_2$

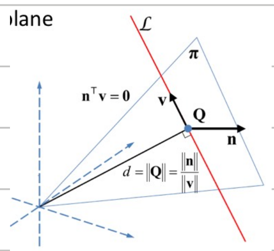
P17 Camera Position Translation : $\underbrace{\begin{bmatrix} t \\ 1 \end{bmatrix}}_{O_c \text{ in } W} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{O_c \text{ in } C}$

P18 Similarity Transformation : $\underbrace{SE(3) : T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}}_{6 \text{ DoF}} \Rightarrow \underbrace{Sim(3) : T_s = \begin{bmatrix} sR & t \\ 0 & 1 \end{bmatrix}}_{7 \text{ DoF}}$ Scale

P20 Plücker coordinates : $\vec{v}$: direction of 3D line $\quad n = Q \times V$

$\vec{n}$: normal of projection plane $\quad \|n\| = d^{*} \|v\|$

$d = \|Q\| = \frac{\|n\|}{\|v\|}$



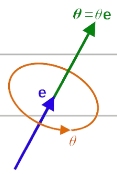
↓

P22 ~~Transformation~~ : $\begin{bmatrix} n_j \\ v_j \end{bmatrix} = \begin{bmatrix} R_{ji} & [t_{ji}]_\times R_{ji} \\ 0 & R_{ji} \end{bmatrix} \begin{bmatrix} n_i \\ v_i \end{bmatrix}$

P24 Euler Angles : Pitch / Roll / Yaw / Intrinsic / Extrinsic Rotation

[illegible] $\quad$ XYZ ~ $\quad$ XYZ stable

P27 : Gimbal Lock : loss one degree



P28 Axis Angles : Angle $\boxed{\theta}$ / Axis unit vector $\boxed{e}$ / Rotation vector $\boxed{\theta e}$ $= [e, \theta]$

P31 Quaternion : 2D: $e^{i\varphi} = \underbrace{\cos\varphi}_{real} + \underbrace{i \sin\varphi}_{imaginary} \Rightarrow$ 3D: $q = (\underbrace{q_0}_{real}, \underbrace{q_1, q_2, q_3}_{imaginary})$ $\begin{cases} q_0 = \cos\frac{\theta}{2} & q_2 = \hat{y} \sin\frac{\theta}{2} \\ q_1 = \hat{x} \sin\frac{\theta}{2} & q_3 = \hat{z} \sin\frac{\theta}{2} \end{cases}$

$q = \underbrace{q_r}_{real} + \underbrace{q_i i + q_j j + q_k k}_{imaginary}$, with $q = [\cos\frac{\theta}{2}, n \sin\frac{\theta}{2}]$

P34 : $p' = qpq^{-1} \Leftrightarrow p' = Rp \Leftarrow$ 3D point : $p = 0 + p_x i + p_y j + p_z k$, with $p = [0, x, y, z] = [0, V]$

P43 3D Representation Methods : SLAM / Point Clouds / Voxel Grid / Implicit Surface / Mesh

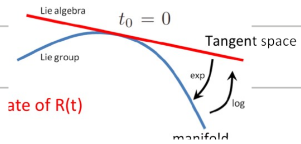
P48 Comparision

	Voxel	Point cloud	Polygon mesh
Memory efficiency	Poor	Not good	Good
Textures	Not good	No	Yes
For neural networks	Easy	Not easy	Not easy

02_ Motion and Scene Representation _02

P03 ~~Group~~ : An algebraic structure of one set plus one operator $\Rightarrow G = (A, \cdot)$

- Closure: $\forall a_1, a_2 \in A,\ a_1 \cdot a_2 \in A.$
- Associative law: $\forall a_1, a_2, a_3 \in A,\ (a_1 \cdot a_2) \cdot a_3 = a_1 \cdot (a_2 \cdot a_3).$
- Identity element: $\exists a_0 \in A,\ \text{s.t. } \forall a \in A,\ a_0 \cdot a = a \cdot a_0 = a.$ "0" for addition, "1" for multiplication
- Inverse: $\forall a \in A,\ \exists a^{-1} \in A,\ st\ a \cdot a^{-1} = a_0.$ x_0 and $-x_0$ for addition, x_0 and $1/x_0$ for multiplication



P04 Common Group : General Linear group : $GL(n)$ / Special Orthogonal group : $SO(n)$ / Special Euclidean group $SE(n)$

P05 Lie Group : a group with continous (smooth) properties // E.g. $SO(n)$ and $SE(n)$

Two matrix can be multiple not plus

P09 Lie Algebra : $so(3) = \{ \phi \in \mathbb{R}^3 \text{ or } \Phi = \phi^\wedge \in \mathbb{R}^{3\times3} \}$ // Skew-symmetric matrix $\Phi = \phi^\wedge = \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix}$

$R = \exp(\phi^\wedge) \Leftrightarrow$ map any vector in $so(3)$ to rotation matrix in $SO(3)$

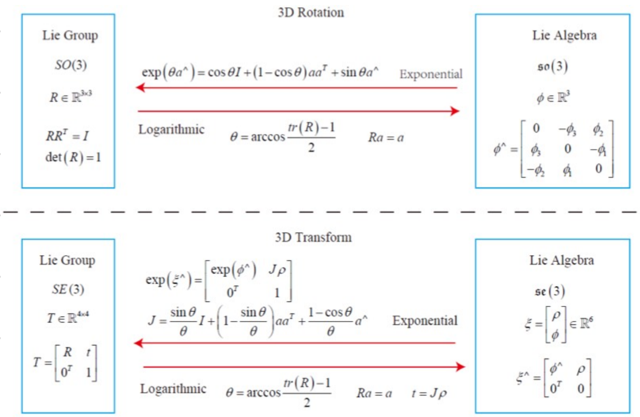
P10 Lie Algebra : $se(3) = \left\{ \xi = \begin{bmatrix} \rho \\ \phi \end{bmatrix} \in \mathbb{R}^6,\ \rho \in \mathbb{R}^3,\ \phi \in so(3),\ \xi^\wedge = \begin{bmatrix} \phi^\wedge & \overset{\text{translation part}}{\rho} \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{4\times4} \right\}$

P11 $so(3) \rightarrow R$: $\exp(\phi^\wedge) = \exp(\underbrace{\theta n^\wedge}_{\text{Axis Angles}}) = \underbrace{\sum_{n=0}^{\infty} \frac{1}{n!} (\theta n^\wedge)^n}_{\text{Tylaor Expansion}} = \underbrace{\cos\theta I + (1-\cos\theta) n n^T + \sin\theta n^\wedge}_{\text{Rodrigues' formula}}$ ⇐ 记住 ☹

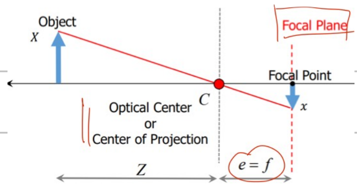
P12 $SO(3) \rightarrow so(3)$: $\phi = \ln(R)^\vee = \left(\sum_{n=0}^{\infty} \frac{(-1)^n}{n+1} (R-I)^{n+1} \right)^\vee$ / $\theta = \arccos\left(\frac{trR - 1}{2} \right)$ / $R\underset{\text{axis}}{n} = 1 \cdot n$

~~P13~~ BCH Formula : $R_1 + R_2 \notin SO(3)$ / $\phi_1 + \phi_2 \in so(3)$

$\ln(\exp(A)\exp(B)) = A + B + \cdots$



03_Image Formation



$$-\frac{x}{f} = \frac{X}{Z}$$

$$x = -\frac{f}{Z}X$$

P07 Pinhole Camera : $-\frac{x}{X} = \frac{f}{Z}$

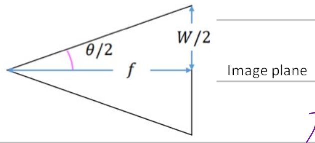
P09 Intersection : vanishing point intersection of parallel 3D lines

↓

P10 : vanishing direction vanishing point and camera center

~~P12~~ ~~FoV~~ : Field of View ⟺ Angular portion

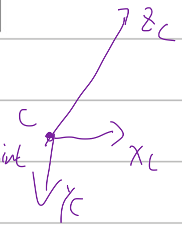
Inversely proportional of the focal length



P15 $\tan\frac{\theta}{2} = \frac{W}{2f} \Rightarrow f = \frac{W}{2}\left[\tan\frac{\theta}{2}\right]^{-1}$ W: image width

← see detail

image center ≠ principal point

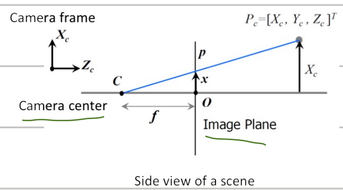


P18 Perspective Projection vs. Parallel Projection

Front plane

P19 Camera Frame → Image Coordinate :

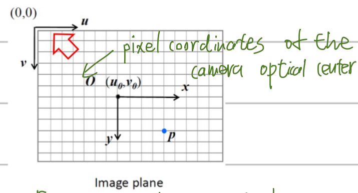
$$\begin{cases} \frac{x}{f} = \frac{X_c}{Z_c} \Rightarrow x = \frac{fX_c}{Z_c} \\ \frac{y}{f} = \frac{Y_c}{Z_c} \Rightarrow y = \frac{fY_c}{Z_c} \end{cases}$$



Side view of a scene

P20 Image Coordinate → Pixel coordinate :

$$\begin{cases} u = u_0 + k_u x \Rightarrow u = u_0 + \frac{k_u f X_c}{Z_c} = u_0 + \frac{\alpha_u X_c}{Z_c} \\ v = v_0 + k_v x \Rightarrow v = v_0 + \frac{k_u f Y_c}{Z_c} = v_0 + \frac{\alpha_v Y_c}{Z_c} \end{cases}$$



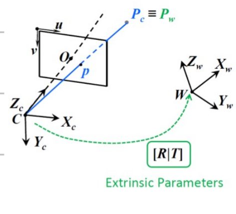
k_u, k_v : pixel conversion factors // α_u, α_v : Focal length in pixels

P22 Intrinsic / Calibration Matrix : $\hat{p} = \lambda\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$

α_u, α_v : Focal length

u_0, u_v : principal point

~~P23~~ ~~Skew factor~~ : today skew factor = 0 ⟺ $\alpha_u = \alpha_v$



P26 World Frame → Camera Frame : $X_c = RX_w + t$

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \left[\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{array}\right]\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \left[\, R \,|\, T \,\right]\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

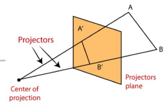
P27 World Frame → Pixel coordinate :

$$\lambda\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_x \\ 0 & \alpha_v & u_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \left[\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{array}\right]\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \underbrace{K[R|T]}_{\text{Projection Matrix }(M)}\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

P31 Line Projection : ① Two step: $A' = [K]A$, $B' = [K]B$ $\Rightarrow$ $\boxed{l = A' \times B'}$

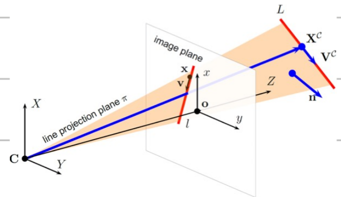
$\uparrow$ 3D vector

Intrinsic Matrix



~~P32~~ : ② ~~One step : $l = Kn$~~

$\uparrow$ 3D vector

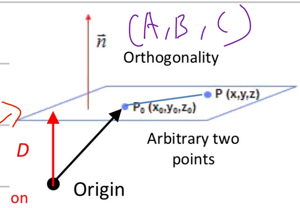


P33 Relationship ··· : Homogenous coordinates of a plane : (A, B, C, D)

Normal of plane — position of plane

$Ax + By + Cz = D$

A point $P\ (x, y, z, 1)$ lies on plane $\Leftrightarrow$ $p^T n = 0$



P34 : Projection plane computed by image line : $\pi_L = \overset{4\times3}{P^T}\, \overset{3\times1}{l_L} \in \mathbb{R}^4$

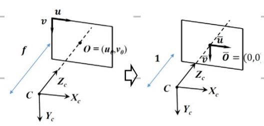
projection matrix

Intersection between 3D line and 3D plane : $D = L\pi \Leftarrow L = \begin{bmatrix} [n]_\times & V \\ -V^T & 0 \end{bmatrix} \Leftarrow \mathcal{L} = (n^T, V^T)^T$

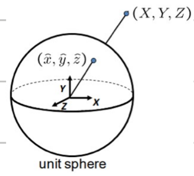
plucker matrix — plucker coordinates

P36 Normalized Image : Focal length equal to 1 unit // origin of the pixel coordinates at the principal point

$$\begin{bmatrix} \bar{u} \\ \bar{v} \\ 1 \end{bmatrix} = K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\alpha} & 0 & -\frac{u_0}{\alpha} \\ 0 & \frac{1}{\alpha} & -\frac{v_0}{\alpha} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{u-u_0}{\alpha} \\ \frac{v-v_0}{\alpha} \\ 1 \end{bmatrix}$$



$$\lambda \begin{bmatrix} \bar{u} \\ \bar{v} \\ 1 \end{bmatrix} = \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix}$$



P44 Spherical Projection : $(\hat{x}, \hat{y}, \hat{z}) = \frac{1}{\sqrt{X^2+Y^2+Z^2}}(X, Y, Z)$

Spherical coordinates: $r = \sqrt{x^2+y^2+z^2}$, $\theta = \tan^{-1}\left(\frac{y}{x}\right)$, $\phi = \cos^{-1}\left(\frac{z}{r}\right)$

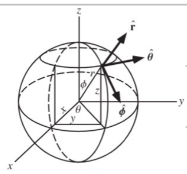
$x = r\cos\theta \cdot \sin\phi$

$y = r\sin\theta \cdot \sin\phi$

$z = r\cos\phi$

Azimuth $\theta \in [0, 2\pi)$

Polar angle $\phi \in [0, \pi)$



P50 Icosahedral representation : extrude the vertices into icosahedron

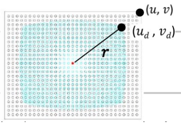
03_Image Formation_02

lens misaligned — light ray bend more near the edges

P02 Image Distortion : Tangential Distortion / Radial Distortion

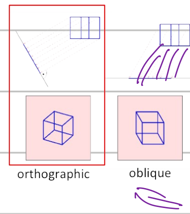
ideal (non-distorted) $(u, v) \rightarrow$ real (distorted) (u_d, v_d)

P05 Quadratic model : $\begin{bmatrix} u_d \\ v_d \end{bmatrix} = \underbrace{(1+k_1 r^2)}_{\text{unknown}} \begin{bmatrix} u-u_0 \\ v-v_0 \end{bmatrix} + \underbrace{\begin{bmatrix} u_0 \\ v_0 \end{bmatrix}}_{\text{principal point}}$ $\quad r^2 = (u-u_0)^2 + (v-v_0)^2$



P14 Orthographic Projection : Projection lines are perpendicular to the image plane

special parallel projection

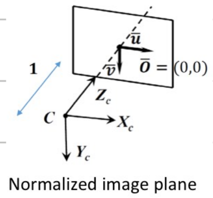


P16 Depth Camera : depth information

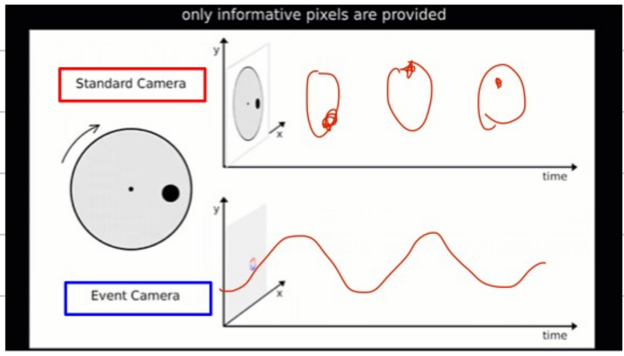
RGBD

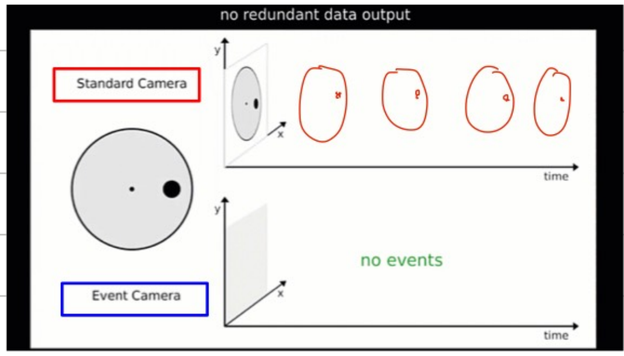
3D - Reconstruction

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



P25 Event Camera : Report only when brightness changes





x

04 _ Camera Calibration

P02 Calibration :

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{K}_{\text{Intrinsic/Calibration matrix}} \underbrace{[R|T]}_{\text{Extrinsic Matrix}} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{K} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

P03 : simultaneously calibration of extrinsic and instrinsic parameters / calibration and localization

Single image

P07 Tsai's Method: Measure 3D position of $n \geq 6$ 3D control points in world frame and their 2D coordinates of projections in the image

P08 DLT :

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K[R|T] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \Rightarrow \lambda \underbrace{\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}}_{\substack{\text{2D projection coordinate} \\ \text{known}}} = \underbrace{\begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\substack{\text{Intrinsic matrix} \\ \text{unknown}}} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}}_{\substack{\text{Extrinsic matrix} \\ \text{unknown}}} \underbrace{\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}}_{\substack{\text{3D points in world frame} \\ \text{known}}} = M \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

composed by points

P14 solving DLT : $\underbrace{Q}_{\substack{2n * 12 \text{ matrix} \\ \text{known}}} \underbrace{M}_{\substack{12\text{-dim vector} \\ \text{unknown}}} = 0$ // $Q\,(2n \times 12)$ should have rank 11 $\Rightarrow$ unique non-zero solution of M

3D-2D provides 2 correspondence $\Rightarrow$ need $\boxed{\text{at least 6 point}}$

$\Longrightarrow$ Least-Squares solution , $\min \|QM\|^2$, subject to $\|M\|^2 = 1$

P16 SVD : $\arg\min_b \|Ab\|_2^2$, subject to $\|b\|_2 = 1$ $\Rightarrow$ $A = U \Sigma V^T$ // optimal solution b^* is the $\boxed{\text{last column of } V}$

= smallest singular value

P18 QR/RQ : $A = RQ$ (Q: orthogonal, R: upper triangular) / $M = \underbrace{\begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{R} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}}_{Q}$

P19 Practical Setup : use more than 6 points and non-coplanar

P20 Zhang's Method : Single image $\Leftrightarrow$ Multiview images // Asume all points are coplaner . $Z_w = 0$

rely on 3D coplaner pointe

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K[R|T] \begin{bmatrix} X_w \\ Y_w \\ 0 \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ 0 \\ 1 \end{bmatrix}$$

$$\Rightarrow \underbrace{\begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix}}_{H} \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix}$$

P26 DLT : $Q \cdot H = 0$ // Q should have rank 8 to have a unique (up to scale) non-trivial solution H

$\underset{\text{Known}}{2n\times 9}$ $\underset{\text{unknown}}{9\times 1}$ Each points provide 2 correspondence $\Leftrightarrow$ $\boxed{4 \text{ non-collinear}}$ points required

↓

P27 SVD : $\arg\min \|Ab\|_2^2$, subject to $\|b\|_2 = 1$ / $A = U\Sigma V$ / b^* is the last column of V

↓

P28 RQ ? : NO RQ, because Q isn't a orthogonal matrix

↓

$$\begin{bmatrix} h_{11}^j & h_{12}^j & h_{13}^j \\ h_{21}^j & h_{22}^j & h_{23}^j \\ h_{31}^j & h_{33}^j & h_{33}^j \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11}^j & r_{12}^j & t_1^j \\ r_{21}^j & r_{22}^j & t_2^j \\ r_{31}^j & r_{32}^j & t_3^j \end{bmatrix}$$

Each view corresponds to a homography

different H^j same K

P29 : Each view has a different H^j, but same K

$\Rightarrow$ must use multiple view

↓

Intrinsic Matrix

P30 K : ① $H = \underbrace{[h_1 \ h_2 \ h_3]}_{\text{known}} = s \underbrace{\boxed{M}[r_1 \ r_2 \ t]}_{\text{unknown}}$ // $r_1 = \lambda M^{-1} h_1$, $r_2 = \lambda M^{-1} h_2$, $t = \lambda M^{-1} h_3$, $\lambda = s^{-1}$ // r_1, r_2 orthogonal: $r_1^T \cdot r_2 = 0 \Rightarrow h_1^T M^{-T} M^{-1} h_2 = 0$

K

$\|r_1\| = \|r_2\| = 1$, $r_1^T r_1 = r_2^T r_2 \Rightarrow h_1^T M^{-T} M^{-1} h_1 = h_2^T M^{-T} M^{-1} h_2$

↓

P33 : ① Assume $B = M^{-T} M^{-1} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix}$ symmetric matrix, estimate 6 elements, e.g. $B_{11} = \frac{1}{f_x^2}$ // $h_i^T B h_j = \underset{6\times 1}{[\ \cdots\ ]} \underset{1\times 6}{\begin{bmatrix} B_{11} \\ B_{12} \\ B_{22} \\ B_{13} \\ B_{23} \\ B_{33} \end{bmatrix}}$

>3 views

↓

P34 SVD : ① stack 2N equations from N views, to yield linear system $Ab = 0$.

Need more than 3 views SVD

↓

P35 [R|T] : ② compute $r_1 = \lambda M^{-1} h_1$, $r_2 = \lambda M^{-1} h_2$, $r_3 = r_1 \times r_2$, $t = \lambda M^{-1} h_3$ subject to $\|r_1\| = \|\lambda M^{-1} h_1\| = 1$ // build $R_i = (r_1, r_2, r_3)$ and project it onto SO(3)

P36 Image Undistortion : Radial Distortion : light rays bend more near the edges of a lens

Tangential Distortion : lens misaligned

$$\underset{CF}{\begin{bmatrix} x \\ y \\ z \end{bmatrix}} = R \underset{WF}{\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}} + t \Rightarrow \begin{aligned} x' &= x/z \\ y' &= y/z \\ u &= f_x \cdot x' + c_x \\ v &= f_y \cdot y' + c_y \end{aligned} \xrightarrow[\text{distortion coefficients}]{\text{Distortion Model}} \begin{aligned} u &= f_x \cdot x'' + c_x \\ v &= f_y \cdot y'' + c_y \end{aligned}$$

↓

P39 Joint Estimation : intrinsic parameters $\rightarrow$ extrinsic parameters $\rightarrow$ use GD, minizie the Reprojection error

distortion coefficients = 0 $\rightarrow$ intrinsic parameters update the intrinsic/extrinsic/distoriton

P40 Reprojection error : Euclidean distance between observed image point and 3D point reprojected on camera frame

$$K, R, T, \text{lens distortion} = \arg\min_{K,R,T,k_i} \sum_{i=1}^{n} \| p^i - \pi(P_w^i, K, R, T, k_i) \|^2$$

P45 Line-based Undistortion :

05 _ Correspondence Estimation _ 1

P02 Matching / Tracking : Template image $T(x) \xrightarrow[W(x,p)]{\text{wrap}}$ Current image $I(W(x,p))$ ← find

P03 Euclidian Transformation : $\begin{cases} x' = x \cdot \cos a_3 - y \cdot \sin a_3 + a_1 \\ y' = x \cdot \sin a_3 - y \cdot \cos a_3 + a_2 \end{cases}$ 1 Rotation, 2 Translation

Task : Find correspondence
Method : Indirect and direct method
Chicken - egg problem ⇔ need constraints

P04 Indirect methods : + cope with large frame to frame motions
− slow , computational cost

① Detect and match features (SIFT)
② Geometric verifiction (RANSAC)
③ Minimize the features distance: $p = \arg\min_p \sum_{i=1}^{N} \| w(x^i, p) - f'^i \|^2$

P06 Direct methods : + All information in the image can be exploited
+ Increasing the camera frame rate reduces computational cost (no RANSAC needed)
− sensitive to frame motion

① Directly processing pixel intensities
② Minimize the intensity distance : $p = \arg\min_p \sum_{x \in T} [I(W(x,p)) - T(x)]^2$ ← SSD

P09 Assumption : Brightness constancy / Temporal consistency / Spatial coherency

P12 KLT : Select features ⇒ Tomasi - Kanade : Choosing best features
Tracking ⇒ Lucas - Kanade : Aligning tracking

P16 Pure Translation : ① $SSD(u,v) = \sum_{x,y \in \Omega} [I_0(x,y) - I_1(x+u, y+v)]^2$ Sum of square of differences, $\approx I_1(x,y) + I_x u + I_y v$

$\Rightarrow SSD(u,v) \approx \sum_{x,y \in \Omega} [I_0(x,y) - I_1(x,y) - I_x u - I_y v]^2 \approx \sum_{x,y \in \Omega} [\Delta I - I_x u - I_y v]^2$ (ΔI: different Intensity)

② Minimize SSD : $\frac{\partial SSD}{\partial u} = 0$ / $\frac{\partial SSD}{\partial v} = 0$ ⇒ $\begin{cases} \sum I_x (\Delta I - I_x u - I_y v) = 0 \\ \sum I_y (\Delta I - I_x u - I_y v) = 0 \end{cases}$

must renber

③ $\underbrace{\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}}_{\text{known}} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{\text{unknown}} = \underbrace{\begin{bmatrix} \sum I_x \Delta I \\ \sum I_y \Delta I \end{bmatrix}}_{\text{known}} \Rightarrow \begin{bmatrix} u \\ v \end{bmatrix} = \underbrace{\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}^{-1}}_{= M^{-1}} \begin{bmatrix} \sum I_x \Delta I \\ \sum I_y \Delta I \end{bmatrix}$

$d(M) \neq 0$

④ $M = \begin{bmatrix} \Sigma I_x I_x & \Sigma I_x I_y \\ \Sigma I_x I_y & \Sigma I_y I_y \end{bmatrix} = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$ // det (M) != 0 ⇒ eigen values should large

⇒ means: a corner/edge

P20 choose Features with large Eigen values

Tracking $\begin{bmatrix} u \\ v \end{bmatrix} \Rightarrow \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \Sigma I_x I_x & \Sigma I_x I_y \\ \Sigma I_x I_y & \Sigma I_y I_y \end{bmatrix}^{-1} \begin{bmatrix} \Sigma I_x \delta I \\ \Sigma I_y \delta I \end{bmatrix}$

P21 General case : ① $SSD(u,v) = \sum_{x \in T} [I(W(x,p)) - T(x)]^2$ General case with wrap

② use Gauss-Newton to get the optimal SSD*

③ Assume : $SSD = \sum_{x \in T} [I(W(x, p + \delta p)) - T(x)]^2$

④ use Taylor-expansion and set derivate to 0

P22 For both cases use first-order approximation

For pure-T ⇒ Partial derivatives

For general ⇒ use Gauss-Newton

P23 Assume : $SSD = \sum_{x \in T} [I(W(x, p + \underbrace{\boxed{\delta p}}_{\text{unknown}})) - T(x)]^2$

P27 Another derivation: Brightness Constancy / constant motion in neighborhood

05_ Correspondence Estimation _02

P04: Feature Detector → Blob : A group of connected pixels share common property

⇒ corner : intersection of two edges / high localization / less distinctive
Blob : group of connected pixels / less localization / more distinctive

P06: → corner Detection : Idea : In corner, image gradient has multiple dominat directions
shifting windows cause large intensity changes.

Moravec's Method : ① compute the SSD of the corresponding pixels in two patches
② ↓ SSD = high similiarity // ↑ SSD = low similiarity
③ smallest SSD = interest measurement
④ interest measurement higher than a threshold ⟺ corner

: $SSD(\Delta x, \Delta y) \approx \sum_{x,y \in \Omega} [I_x(x,y)\Delta x + I_y(x,y)\Delta y]^2$ ⇐

⇒ $[\Delta x \ \ \Delta y] \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$

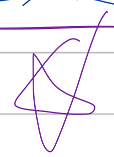
manually select

Eigenvalues

$$M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

Eigenvectors

I_x and I_y: Directional derivatives



conclusion : $\lambda_1 = 0, \lambda_2 > 0$ ⟺ edge
$\lambda_1 = 0, \lambda_2 = 0$ ⟺ flat region
$\lambda_1 > 0, \lambda_2 > 0$ ⟺ corner

Summary : Above method is called Harris detector
It's $\boxed{\text{not scale invariant !!}}$

P14 Feature Descriptor → Brute force matching : check all the points $O(N^2)$ N^2 ⇒ comare $\underset{\text{left}}{N} \times \underset{\text{right}}{N}$
choose the minimum distance / closest description
⇒ problem : falsch matches with good scores
⇒ solution : ratio of distance / $\frac{d_1}{d_2}$ < Threshold (0.8)

dist / d2nd

P16 Distance Function : $SSD = \sum_{u=-k}^{k} \sum_{v=-k}^{k} (H(u,v) - F(u,v))^2$ ≥ 0
$SAD = \sum_{u=-k}^{k} \sum_{v=-k}^{k} |H(u,v) - F(u,v)|$

Similarity measurement : $NCC \in [-1, 1]$ ... $= 1$ if $H = F$
$ZSSD = \sum_{u=-k}^{k} \sum_{v=-k}^{k} [(H(u,v) - \mu_H) - (F(u,v) - \mu_F)]^2$

zero-mean

P18 Descriptor : "description" of the pixel informations around a feature

① ~~Ø~~

P20 Patch Warping Method : ① determine Scale, rotation, view point change
② wrap each patch into a canonical patch
③ establish patch correspondence

P21 Scale: problem: Two images with different scales / Two patches with same size
solution: resize patches with different sizes / time-consuming / $N^2 S$
better solution: introduce scale invariant function extremum
Sharp intensity changes / multiple extrema
Function: kernel can highlights the edges : f = kernel * Image
Apply Laplace operator / Laplacian of Gaussian = LoG (x, y, σ) $\sigma = r/\sqrt{2}$

P34 Rotation: patch orientation: gradient vectors of each pixel → histogram → dominant orientation

P36 View of point : Affine wraping

P38 Disadvantage : small errors affect the matching score
computationally expensive

05_Correspondence Estimation_03

P03 Different descriptions : Patch descriptor / Census descriptor ⇐ vector with integer or float

patch descriptor

P12 Brute force → Drawbacks : ① Scale determination depends on tentative matching

② Fix the scale, but not the optimal

Improvements : If know the scale of patch : $O(N^2 s) \to O(N^2)$

problem : How to determine the scale of patch first ?

P14 Automatic scale determination : automatically assign each patch by its own size

⟺ based on a single image

independent of tentative matching

P15 use brute force to match known scales

P16 Rotating Patch Descriptor : ① Harris detector is rotation invariant

Eigen vectors of M = directions changes of SSD

P17 ② Compute gradient vector at each pixel of the patch

Histogram → weighted by gradient magnitudes

Local maxima > threshold is dominant orientation

P18 Blob with scale : Blob : A group of connected pixels share common property

Aim : Given a single image, detect blob, automatically assign scales

P20 Blob centers = point correspondence, not so accurate

P24 How to detect a blob and optimal scale σ ⟺ $\boxed{\sigma = r/\sqrt{2}}$

Idea : Try a set of candidates (LoG)

此处应是比较同一pixel不同尺度的的response

Apply a kernel on a single image : $\underset{\text{response}}{f} = \underset{\text{LoG w.r.t. } \sigma}{\text{kernel}} * \underset{\text{patch}}{\text{image}}$

↳ Maximum response = optimal $\sigma = r/\sqrt{2}$

P28 NMS : Two near pixel reach the maximum response with same σ ;

此处应是比较不同pixel的相同scaled极值

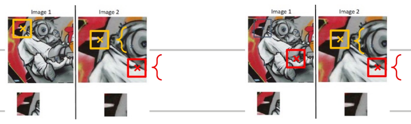
use NMS, compare Extremum and choose the heiger pixel

P29 Summary : Build Laplacian → with inital scale → iterate n

LoG → increase scale by k

↓ NMS

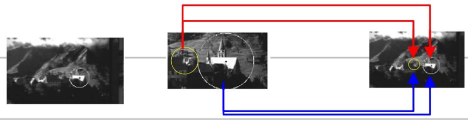
P31 Complexity : ✓ "Straightforward scale search" vs. "automatic scale determination"



Simultaneously find the optimal patch correspondence and optimal scale

Complexity: $N^2 \bullet S$ = scale of patch

Brute force matching with **straightforward** scale search (one-step method)



First step: Scale determination

每个blob 都应有 自己的 s

Second step: brute force matching based on rescaled patches

NS

Complexity: $S + S + N^2$

Brute force matching with **automatic** scale determination (two-step method)

P32 SIFT: Problem : Patch descriptor-based method is sensitive to noise / LoG is inefficient

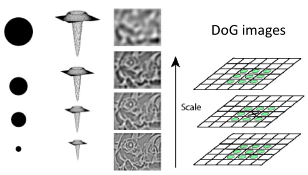
→ Instead of using SSD, compare their associated vector descriptors

→ Instead of using LoG, use $DoG(x,y) = G_{k\sigma}(x,y) - G_{\sigma}(x,y)$ at different scales

P35 key point Extraction : Gaussian kernel → maintains image size / blur the image → DoG

P38 SIFT key points : Compare each pixel with 8 + 9 + 9 neighbors

If pixel is extremum then it is SIFT feature

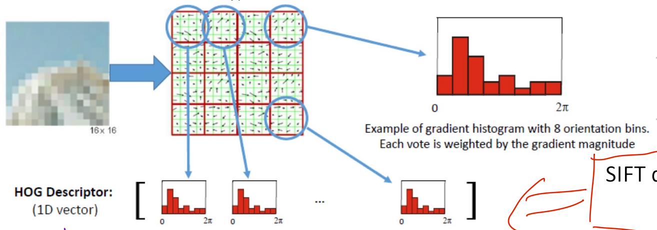


Same filter size 3x3 with different scale σ

P42 Descriptor Compution : Compute histogram of oriented gradients

⇒ concate all histogram into 1D vector

Derotate ⇒ 16*16 pixels



Example of gradient histogram with 8 orientation bins.
Each vote is weighted by the gradient magnitude

SIFT descriptor: 4 × 4 × 8 = 128 values

16 cells 8 bins

1 X 128

P43 output : Location → 2D vector

Scale → 1 scale value

Orientation → 1 scale value = angle of patch domain diretion

Descriptor → 4x4x8 = 128 element 1-D vector